Communication Patterns in Automotive Systems

Ruben Broux, Elena Lisova, Saad Mubeen

Mälardalen University Sweden,
firstname.lastname@mdu.se

Abstract. In the last decade, electronics and software have replaced many mechanical components in vehicles at an unprecedented rate. New emerging technologies have found their way into the vehicular domain, like for instance, ADAS systems. This change brings some particular challenges with it in terms of functionalities, safety and security. Many vehicle distributed functions require hard real-time and secure communication. Therefore, the electrical and electronic (E/E architectures) architectures are in a continuously adapting trend to meet the new standards. The adaptation from a distributed to a domain-centralized architecture is already present. It is crucial to facilitate reuse of system architectural solutions in order to make system development more efficient. Therefore, we propose the use of communication architectural patterns. We present a method to map communication patterns on a certain layer of abstraction. The method has been evaluated through several industrial use cases. Furthermore, this work sets precedence for future research and development, as well as future applications of the method.

1 Introduction

The automotive industry is a rapidly changing industry. Electronics have been replacing mechanical components at an unprecedented rate [4,9]. Original Equipment Manufacturers (OEM) keep adding new functionalities to their vehicles, which are all controlled by on-board embedded systems. Those embedded systems are linked to each other via different networks within the Electrical and Electronic (E/E) architectures. They enable aspects such as safety, security and functionality of the vehicle. There is no standard E/E architecture accepted and shared by vehicle manufacturers however they are following similar trends in development. There is a shared consensus on three common E/E architectures. Namely the distributed, domain-based, and vehicle-centralized architectures. Typically, the distributed architecture is applied in most of the vehicles in today's market. Moreover, those vehicles may contain up to 100 on-board dedicated ECUs (i.e., Electronic Control Unit) [9]. This complex architecture is no longer capable of meeting the new requirements from innovations such as Advanced Driving Assistance Systems (ADAS). Typically, those requirements are timing and security related [23,24]. But also bandwidth requirements can be set, since the flow of data has drastically increased in a vehicle (e.g., radar sensors and cameras). In addition, this architecture makes the system development less

efficient.

Therefore, one may see a gradual shift to the domain-centralized architecture. This architecture contains less but more powerful ECUs. A direct effect of this will be a less expensive E/E architecture, since there will be less ECUs [40]. Communication over the network will decrease while communication in the domain controllers will increase. Therefore, the domain architecture will be faster and more capable of meeting the new requirements.

Future systems will become very complex. To mitigate the complexity of the vehicle, future architectures will be focused on a centralized architecture. In addition, this would decrease the amount of ECUs and the remaining ECUs will be very powerful. Moreover, they will be capable of handling for instance, visualization techniques and artificial intelligence. A thorough search of the relevant literature yielded that so far no commercial implementations have been made with a centralized architecture.

It is in the industries interest to speed up the design process of an E/E architecture and allocation of specific functionalities into architecture assuring their proper support. Therefore, the focus of the report is on finding and constructing communication patterns. To get there, a prototype of the method is designed in parallel with the communication patterns. This is done by the use of industrial use cases. Every pattern is derived from a use case, and later on abstracted so it is applicable to more communication systems. The use cases are considered from the segment of construction equipment vehicles and provided by Volvo Construction Equipment.

The problem we are facing in this study is the following. There is a need for an evaluation on various automotive E/E architectures through communication patterns. This need is caused by the fact that, to the author's best knowledge, this subject has never been researched before. The patterns will ease the reuse of system architectural solutions in order to make the system development more efficient. Furthermore, system properties such as security and safety have to be supported by the patterns. Also, the patterns should be on a certain level of abstraction so they are applicable to other communication systems as well. From this problem statement, we can derive our research questions for this report as follows:

- **RQ1:** How to provide classification for communication patterns used in distributed embedded systems in vehicles?
- **RQ2:** How to allocate functionalities in an E/E architecture with the use of the classification?

2 Background

2.1 Technical terms

Signal / Service Oriented Communication Systems A signal-oriented communication systems has all of its communication design exclusively statically specified. Typically, data is published as broadcast disregarding if any subscriber is present. However, the subscription from the client to the publisher has to be made statically. On the other hand, a service-oriented architecture will use a more publish-subscribe or request-response manner. They are driven by the clients and do not require a static subscription.

The paradigm of signal-oriented communication will increasingly be replaced by service-oriented communication. One of the reasons for this change is the increase in capability for updating and upgrading the system [34].

This will bring additional problems for adapting signal-oriented systems to a service-oriented design. Moreover, service-oriented systems will set new requirements and challenges to the automotive security. New protocols, integration of third-party services, and standardized software components will all have their introduction in a service-oriented system, which were not in the scope for security concepts in the signal-oriented architectures [34].

SOME/IP Service-Oriented Middleware over IP (i.e., SOME/IP) is a lightweight protocol to facilitate inter process/device connection. It uses automotive Ethernet as communication medium. Furthermore, it is an automotive specific protocol that provides service-based communication. These services can exist out of events, methods and fields.

- Event: provides data which is sent cyclically or on request from provider to the subscriber.
- Methods: provide the possibility to the subscriber to issue remote procedure calls which are executed on provider side.
- Fields: are used for additional information (e.g., a notifier which sends data on change from the provider to the subscribers).

One of the advantages of SOME/IP is that only data required by a subscriber is send from publisher to subscriber. The traditional way in a signal-oriented architecture would be that all the data is broadcasted in between certain ECUs. Furthermore, SOME/IP was designed to be a middle ware which makes the existing traditional ECUs compatible with the new protocols and technologies [14]. In addition, SOME/IP works over TCP and UDP protocols. Therefore, SOME/IP can not be implemented when using CAN protocol.

Safety standard ISO 13849 The safety standard ISO 13849 provides safety requirements and guidance on the principles for the design and integration of safety-related parts of control systems. The standard is used by Volvo CE instead of the well-known ISO 26262 ASIL levels used for road vehicles. ISO 13849 works

with Performance Levels (PL). The ISO 13849 levels map to the ISO 26262 levels except the lowest level of ISO 13849. Moreover, the PL b level which is second lowest matches with ASIL A level. The highest PL level (i.e., level e) concurs with ASIL D level.

Furthermore, the performance levels in ISO 13849 exist out of several aspects, among which the following is the most notable:

The probability of dangerous failure per hour (PFHd). This parameter expresses the average probability of a dangerous failure happening in one-hour time period. Table 1 gives the relationship between the PL and the PFHd [31].

PL	PFH_d
a	$\geqslant 10^{-5}$ to $< 10^{-4}$
b	$\geqslant 3 \cdot 10^{-6}$ to $< 10^{-5}$
c	$\geqslant 10^{-6}$ to $< 3 \cdot 10^{-6}$
d	$\geqslant 10^{-7}$ to $< 10^{-6}$
e	$\geqslant 10^{-8}$ to $< 10^{-7}$

Fig. 1. Relationship between PL and PFHd [31]

Intrusion Detection System Two types of intrusion detection systems can be distinguished namely: Specification-based detection and Anomaly-based detection [19]. Typically, this is used in Controller Area Network (i.e., CAN) based networks. Specification-based detection uses one detector in every ECU. This detector will investigate incoming and outgoing traffic. Based on the network traffic, a typical communication behaviour table for that specific ECU can be developed. Depending on the behaviour, malicious network traffic can be highlighted and reported to for example the driver of the vehicle. Anomaly-based detection will only use 1 detector which is connected with the network. This detector will investigate all the traffic which occurs on the network and make a list with the most common send messages. In case a deviation from the default traffic is observed, a warning for possibility of malicious content could be issued.

Honeypot A honeypot can be seen as a virtual copy of the in-vehicle network. It will copy its parameters and architecture. As such, it aims on tricking the attacker, so he assumes he is gathering data from the vehicle while it is data from the honeypot [39]. A prerequisite is that the honeypot most be realistic, else the attacker may realize he is not attacking the intended network. It is

suggested that the honeypot is attached to the wireless gateway in the vehicle. The data collected in the honeypot can be send over and analysed. Such that attackers techniques, attacker behavior, and attack trends can be exploited. This can result in the vehicle being less prone for future cyber-attacks.

2.2 Architectures

Distributed Distributed E/E architectures represent the traditional automotive architectures which have been employed for most of the vehicles in today's market. In this architecture, several ECUs are connected to each other over different networks. Every ECU is dedicated to one or more functions like for example an ECU for engine management, or an ECU for brake handling. Therefore, there is little to no interaction in between the ECUs. The architecture is very modular since each ECU has its own function or functions. Furthermore, software and hardware have a high coupling with each other which often leads to the problem of vendor lock-in. AUTOSAR [2] is mainly developed to decouple the software from the hardware. Moreover, to promote reuse, it standardises basic system functions and functional interfaces. This pushes the shift from an ECU-based paradigm to a function-based system design.

Domain Domain centralised E/E architectures represent the present of vehicular architectures. Contrary to the distributed E/E architecture, they focus on scalability and maintainability. A direct consequence is the increase in robustness and decrease in complexity and manufacturing costs. Commonly used domains in automotive industry are [21]:

- Chassis domain: for controlling of the vehicle.
- Powertrain domain: for controlling engine and batteries.
- Body electronics domain: for controlling of climate controls and power windows.
- Advanced Driver-Assistance Systems (i.e., ADAS) domain: for controlling of components within the assistance systems.
- Infotainment domain: for controlling of displays and plug-ins for operators smartphone.

Typically, this architecture exists out of one domain controller per domain. This domain controller acts like a gateway as well. Moreover, the ECUs inside one domain can directly communicate with each other. If an ECU needs to communicate with an ECU in another domain, the communication will happen through the domain controller. In addition, this architecture offers a connection with the internet as well. This will mainly happen over a secure gateway and a wireless protocol like for example Long Term Evolution (i.e., LTE). Long Term Evolution, better known as 4G LTE is a standard for wireless data transmission. Typically, this is used in mobile cellphones for downloading of music, websites and videos.

Centralized Vehicle centralized E/E architectures represent the future of vehicle architectures. While domain-centralized architectures tend to move towards the direction of domains, centralized architecture moves towards the direction of services.

This architecture may exist out of a single master vehicle control computer or computing cluster. The vehicle is physically partitioned into zones and a single ECU is assigned to each zone. These ECUs only act as a gateway between the central computer and the zone. The central computer may exist out of a High-Performance Computing (i.e., HPC) server. This will act as the actual brain of the future vehicles [10]. This opens up a new dimension for emerging technologies such as artificial intelligence, neural networks, cloud, and over-the-air (OTA) updates [8]. However, this architecture brings considerable risk potentials related to security and safety as well. Moreover, vehicles will be exposed to the Internet and are therefore prone to installation of malicious software. Subsequently, the software can compromise the vehicle safety and security.

For now, the domain-based architecture will stay the trend. The centralized is an idea for future works. Moreover, in the current architecture there are already powerful multi-core hardware devices. Also, virtualization techniques are developed which makes it possible to have different OS in different SW partitions. This enables the execution of functionalities with different criticality levels.

2.3 Communication Protocols

CAN CAN (Controller Area Network) [17] was introduced by R. Bosch in 1983. It has data rates up to 1 Mbit/s throughout a twisted pair cable of max length 25 - 40m. It uses auto arbitration and has low overhead. The frame exists out of 64 bits. CAN network has been widely deployed on a large number of vehicles since a variety of embedded chipsets are able to communicate on it. The CAN defines data link and physical layer (i.e., no higher level protocols). Furthermore, security was never considered because the network was expected to only exist out of trusted nodes. CAN uses carrier sense multiple access arbitration with a collision avoidance mechanism based on the ID field of the message.

A node would lose its ID if encryption would be used. Therefore, it is hard to encrypt the physical layer. The ID defines the priority of the node which can cause high priority messages to spam the bus (i.e., DOS attack) [19]. Besides it is possible to broadcast transmission and no authentication is required for new nodes to join. On top of that, CAN is a event triggered protocol which can cause latency's (depending on the priority of the node) in transmission [38]. Nevertheless, more deterministic behaviour can be obtained with the more recent Time Triggered CAN (TTCAN) [20].

CAN XL and CAN FD In the newer CAN XL [3], the speed is increased to 10 Mbit/s. The data part of the frame can vary between 1 to 2048 bytes [12]. Because of this reason, it is easier to implement higher layer security precautions (e.g., tunneling TCP/IP packets over CAN XL). But also, it would allow

the use of SOME/IP. Moreover, there is a possibility to route Ethernet frames on a CAN XL protocol [13]. Furthermore, it is also compatible with CAN FD [1].

The primary difference between classic CAN and CAN Flexible Datarate (CAN FD) is the data-rate. Moreover, CAN FD is able to dynamically switch to a different data-rate. This means data can be transmitted faster than the default 1 Mbit/s for regular CAN. Furthermore, CAN FD can extend its data field. For regular CAN the data-field exists out of 8 bytes. For CAN FD this can be extended to 64 bytes [32].

LIN Due to economical factors LIN was introduced. LIN intended to keep the system simple enough so implementation costs would be reduced to a minimum [33]. Typically, LIN is used for sensor and actuators in the comfort domain of vehicles (e.g., climate control, window motors and windshield wipers).

Local Interconnect Network (LIN) Consortium was founded in the late 1990s. The consortium exist out of automakers Audi, BMW, DaimlerChrysler, Volkswagen and Volvo as well as Volcano Communication technologies and Motorola's expertise in automotive semiconductor hardware products [33]. LIN 1.3 was released to the public in late 2002.

LIN network is implemented using a single wire, which generally means higher values in EME (electromagnetic emission) compared to differential twisted pair wire implementations such as CAN. Therefore, lower bandwidths are used such as 20kbit/s. But also, LIN uses a cyclical-redundancy-check feature to ensure the data corrupted by noise in the wiring is detected. Furthermore, LIN uses a simple checksum, in two optional forms, which is easy to implement in software on a microcontroller. Typically, a LIN network exists out of 1 master and one to fifteen slaves.

Every LIN message has an ID-field. This field exists out of 8 bits in where the 2 last bits represent parity bits. The slave nodes on the network are addressed by the ID-field. Moreover, the nodes do not have a physical address like is typical the case in communication networks (e.g., MAC-address). Instead, they use a pre-programmed list of valid IDs which is saved in their memory. They use this to filter out which messages to respond to. Because of this, the ID send by the master can have different meanings. The ID can be for a single slave (i.e., unicast) or more slaves (i.e., multicast) or all slaves in the network (i.e., broadcast). Since only the master can initiate the communication, there is no need of any collision detection system. The data field is of variable length (e.g., 2, 4 and 8 bytes). LIN does not support any cryptographic primitive. Sending malicious sleep frames can possible disable the network [19].

The master of a LIN-network handles all transmissions of frames according to schedules. Each frame takes up a certain amount of time in the schedule and is called a frame-delay. The schedules are built upon the frame-delay times. Furthermore, the LIN-specification supports the use of multiple schedules. A up-down window button in a car, for instance, could issue an extra schedule. This extra schedule would prioritize the messages sent to and from the window-lift

motor in order to cut down the response-time for when the button is released. This is of great importance since one would not like a delayed window response (e.g., fingers could be pinched).

Despite the previous discussed characteristic regarding prioritizing certain messages in a LIN-network. LIN is not suitable for real-time applications. Moreover, LIN is prone to signal latency's and jitter. Schedule latency's and Transmission latency's are common for LIN [35].

FlexRay The FlexRay protocol [18] has been on the market since 2009 and is designed to be faster and more reliable then CAN. It is able to support 3 different topologies (star, bus and mixed) and fault tolerant operation. Furthermore, it supports data rates up to 20 MB/s and data is transported over 2 or 4 unshielded twisted pair cables. In case it uses 2 unshielded twisted pair cables it is able of offering enhanced fault tolerant communication. Also, it is a time triggered protocol which makes it more reliable and suitable for applications like for instance, brake by wire. FlexRay defines the specifics of a data-link layer independent of a physical layer. It uses Time Division Multiple Access scheme which provides a designated time slot for every node. Since FlexRay only defines the data-link layer, it is still required to implement the application layer. The application layer should assure the security of the protocol and this is missing in the FlexRay protocol [30]. FlexRay offers a CRC check which assures some form of data integrity. However, there are no assurances against confidentiality, authentication and freshness of data [30]. Researches have been able to turn on the braking lights while a vehicle was driving at a velocity of 113mph without applying force on the brake paddle (i.e., spoof attack) [30].

Automotive Ethernet Currently, default Ethernet is used in vehicles mainly for diagnostic access. Flashing a firmware update of 81MB would take CAN around 10 hours, while flashing an update of 1GB would take Ethernet 20 minutes [37]. Ethernet is the perfect solution for high bandwidth applications (e.g., radars, cameras and ultrasonic sensors). Another advantage is the cabling, UTP only uses 1 single twisted pair. Ethernet does provide assurance for authentication and integrity [7] since it uses TCP/IP protocol. On the other hand, default Ethernet does not provide real-time functionality, it is an event triggered protocol. Similar like CAN latency's can occur on the network which makes it not suitable for application such as drive by wire.

BroadR-Reach Ethernet is an Ethernet physical layer standard designed for the automotive industry. BroadR-Reach provides a full-duplex operation at 100Mbps data rate over a single twisted pair of wires. Typically, BroadR-Reach Ethernet is used for infotainment and ADAS applications. However, it is recommended to use BroadR-Reach only for passive ADAS sysems (i.e., ADAS system which only warns the driver). Also, it is more cost-effective than MOST or LVDS. Moreover, the cables of BroadR-Reach Ethernet are thinner than LVDS cables.

Audio Video Bridging Ethernet (i.e., AVB Ethernet) is the predecessor of TTEthernet. AVB Ethernet provides a set of protocols to manage the network time for supporting synchronized operations. For seven hops within the network, AVB guarantees a fixed upper bound latency. Moreover, two Stream Reservation classes are defined. A class A that provides a maximum latency of 2ms and a class B that provides a maximum latency of 50ms [9].

Time-Sensitive Networking (i.e., TSN) is a family of standards which offers reliability, determinism and time synchronization to safety-critical automotive communications over Ethernet links. The TSN standards build up-on the previous work done within the IEEE 802.1 Audio Video Bridging (AVB). Moreover, TSN provides standards for precise time synchronization, deterministic communications, ultra-low latency, zero congestion loss, reliability and fault-tolerance [5].

Time Triggered Ethernet (i.e., TTE) is an Ethernet protocol for real-time communication. It is designed to allow for the coexistence of time triggered real time, synchronized communication with lower priority event triggered messages over Ethernet.

It defines a fault-tolerant synchronization strategy. This will assure synchronized time in distributed systems. In TT Ethernet packets are sent over the network at predefined times and take precedence over lower priority packets. Moreover, lower priority messages are interrupted and stored in a switch buffer to allow the TT messages to take priority [38].

MOST Media Oriented Systems Transport (i.e., MOST) was developed to primarily support multimedia transport. It uses an optical physical layer with data rates up to 150 Mbps (for MOST150) [11]. As transport medium it uses plastic optical fiber, which is sensitive to bends in the wire. Hence, it is harder to built into vehicles. The topology is mainly a logical ring structure which transmits data from one device to another. It is also possible to use a combined ring and start topology. MOST Cooperation publishes most specifications [11] but it lacks details related to security on the data link layer.

LVDS Low-voltage differential signaling (i.e., LVDS) is a protocol used in automotive industries for transmission of video signal from cameras. It delivers high datarates up to 655Mbps [16], low interference radiation and simple cabling. LVDS was introduced in 1994 and is used outside the automotive industry as well (e.g., LCD-TVs, industrial cameras and computers).

2.4 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.

In other words, a pattern can be used as a reusable solution to a commonly occurring problem.

In the case of communication patterns in the automotive domain it will be a template based on the E/E architectures. The main focus is facilitating reuse of architectural solutions in order to make system development more efficient. Therefore, a communication pattern will be divided up in decompositions. As a result it will be more convenient to characterize a communication pattern based on its decomposition.

3 Proposed Method for Identifying Communication Patterns

3.1 Performed Iterations

In the first iteration an in-depth study in the form of a literature review has been performed. To this end the following digital libraries were accessed: IEEE Explore, Springer Link, ResearchGate and ACM DL. Forward and backwards snowballing have been performed to acquire as much relevant information as possible. Besides, information was also collected on web pages and posts of OEM's, suppliers and Google Scholar. In the second iteration the principles about the decomposition of the pattern were discussed. To this end there were no examples of communication patterns. Therefore, before reaching the feature diagram, other tables/trees were first set up. A first idea was a hierarchical tree, which was followed by a table. The final result was a feature diagram.

Subsequently, decompositions of a communication pattern were discussed in detail. Accordingly, recommendations were given to the patterns as well.

Initial Analysis In order to start reasoning over any kind of communication patterns it was necessary to gain knowledge of the system first. As explained previously, relevant information was acquired through accessing digital libraries. Firstly, the concepts of distributed, domain and centralized E/E architectures were examined. Discussions were held about the scope of the report. Boundaries and limitations of the report were set. We set the boundaries on the three generic E/E architectures (i.e., distributed, domain and centralized). Within the domains we would focus on the intra-vehicle communication. More specifically on the wired intra-vehicle communication. Secondly, mind-maps were created in order to keep track of all the new information that was acquired.

One would gain better knowledge of a system by working with it in a hands-on manner. Therefore, we started to link the differences between distributed and centralized to a distributed home automation system which was build by one of the researchers. This system existed out of 6 nodes and each node existed out of an LED and a motion sensor. At the very first, every node was responsible for their dedicated LED. Later on, the system requirements changed. Namely, a motion sensor of an adjacent node need to turn on a LED in another node.

This caused a lot of overhead because the data had to be exported to the other node. Subsequently, a centralized server was build on a RaspBerry PI to which the nodes were connected to in a publish-subscribe manner. Every node would send its motion input to the server and the server would take care of all the functionalities. This adaptation had a vast influence on the maintainability of the system. One would not have to export any data anymore, and every functionality could be programmed directly in the central server. Though, no constraints were present in the system (e.g., timing, safety and memory) like one would typically see in automotive systems. It still reflects the ease of a centralized system. Figure 2 displays the first mind-map.

- Technical terms: these terms were collected throughout the research in the literature. We decided afterwards which one were valuable to us and refined those using the existing literature.
- Evaluation meterics were defined, for instance, cost in terms of required material and complexity. Also development time is including in the costs which means the costs can be expressed in financial terms. Also, bus load was included in this branch which reflects on some characteristics of the used protocols such as for instance, worst-case response time.
- System requirements: this would represent the requirements given by Volvo CE. A first division was made in security, safety and financial requirements.
- Protocols: the most common protocols in the automotive domain were discussed and also some metrics were applied.
- Architectures: the three generic architectures were discussed.

Once this information was acquired, a new mind-map was created for the communication patterns. In here our first idea of a possible classification is displayed. As can be seen in figure 3, we split up the patterns in 4 classes, namely: security, critical, safety and infotainment patterns. Metrics were also applied such as a trigger, purpose, direction and communication mechanism. Moreover, the communication mechanism would describe what protocol would be used in the pattern.

Also, patterns in Autosar classic and adaptive were examined. The most common ones are displayed in the most lower branch.

Furthermore, some security precautions were discussed which were found in the literature. The security precautions can be found in the upper branch in figure 3. A more detailed explanation about the honeypot and intrusion detection system can be found in section 2.

Then, we mapped functionalities of a pattern in a hierarchical way. A hierarchical tree was build with different levels. The first iteration of the tree can bee seen in figure 4. As explained earlier on, the focus of the report would be in particular the wired intra-domain communications. Therefore, there is no elaboration on the wireless side of the tree. Also, when selecting the inter vehicle branch one can typically not choose the wired communication side since these kind of communication does not exist. A simple example is provided here namely the Tire Pressure Monitoring System (TPMS).

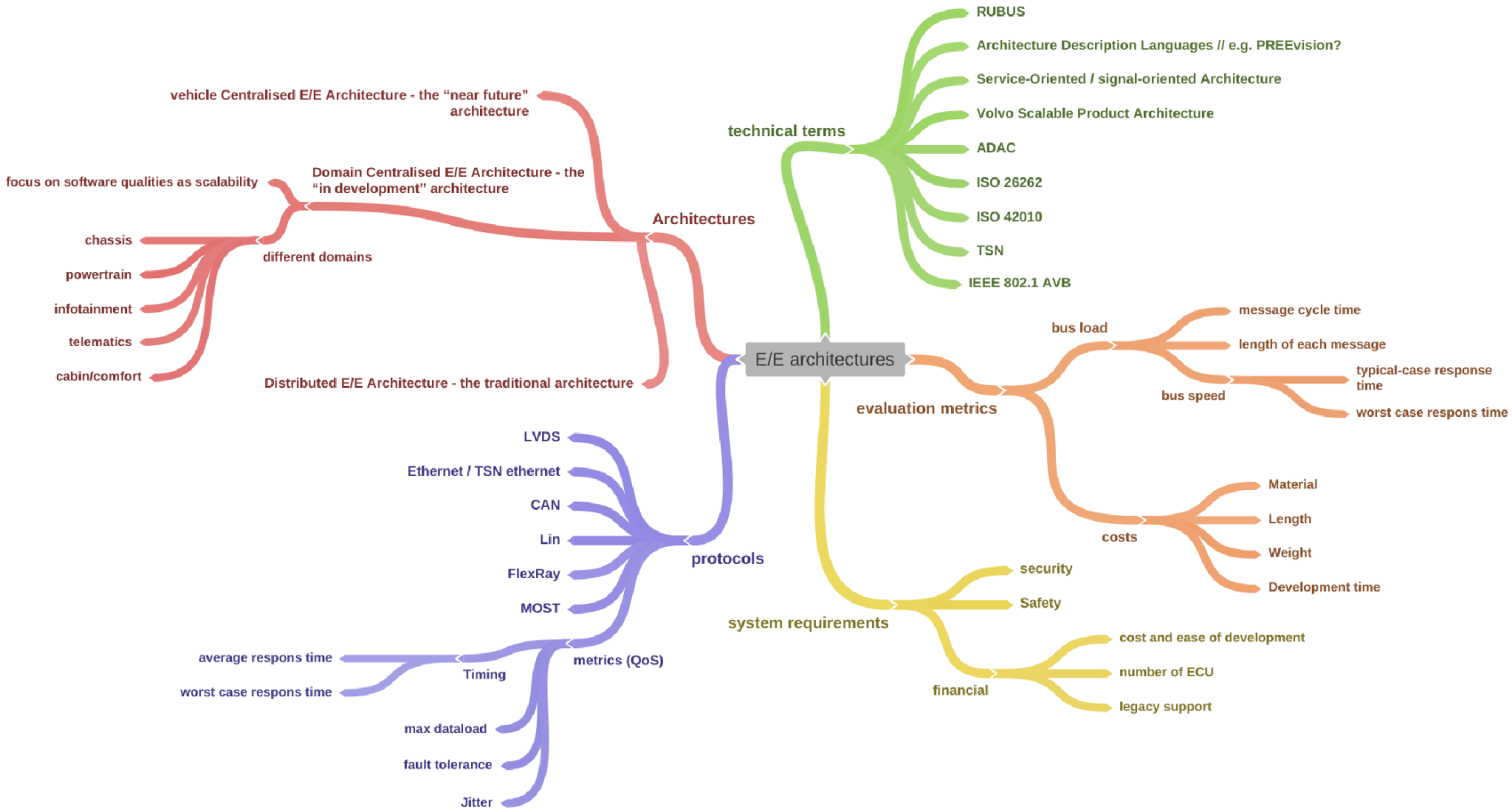

Fig. 2. Mindmap 1

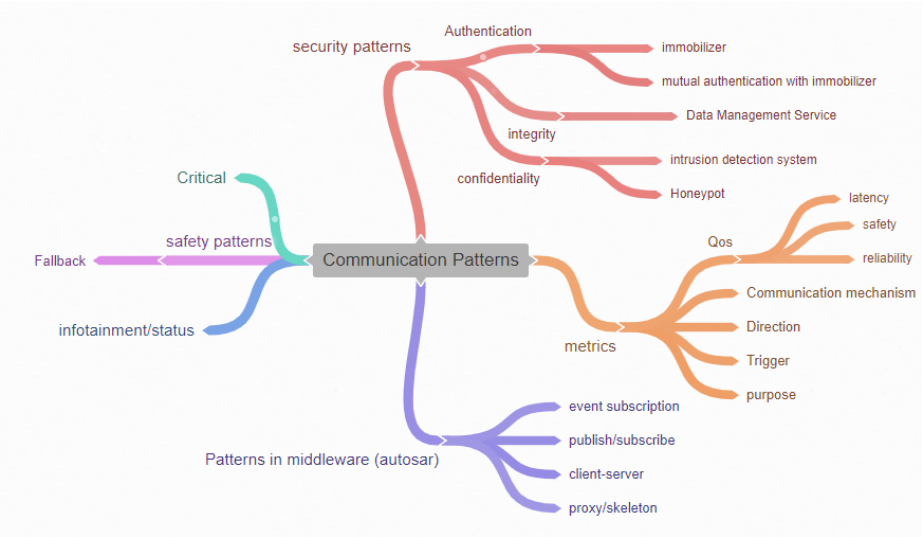

Fig. 3. Mindmap 2

The different levels of the tree are as follows:

- level 1: Intra or inter vehicle communication.
- level 2: Wired (in case of intra vehicle communication) or wireless communication.
- level 3: end-to-end or broadcast messages.
- level 4: time-triggered or event-triggered applications.
- level 5: secured or not secured communication in the terms of encryption.
- level 6: service or signal oriented applications.

As this tree was discussed some relevant remarks were made. No unicast or multicast messages were included in this tree. Also, no timing constraints were displayed in the tree. The name end-to-end could be interpreted in different ways. Therefore, a second tree was designed.

Figure 5 displays the second version of the hierarchical tree. Since every branch is a copy of each other only the broadcast branch has been fully written out. This has been done for the purpose of a better overview.

In the second version of the tree, the missing data transmission patterns were added (i.e., unicast and multicast). Also, we split up time constraints in critical and time triggered. A critical message would get the lowest latency possible (e.g., emergency braking). While a time triggered message is a message which is triggered in a periodical way. On the other side we have 'no time constraints', which refers to event-based triggered messages (e.g., the activation of the air conditioning in a vehicle). Also the name end-to-end was changed to point-to-point to avoid any confusions. Furthermore, the concept of encryption was

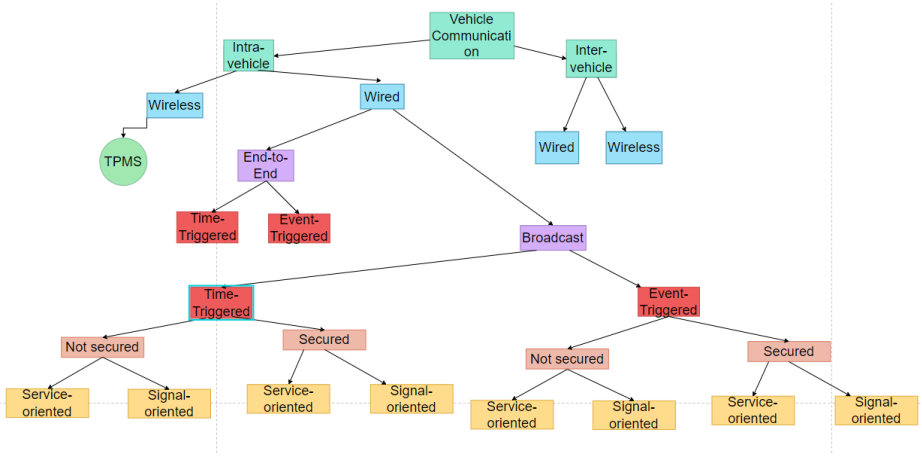

Fig. 4. Pattern Tree 1

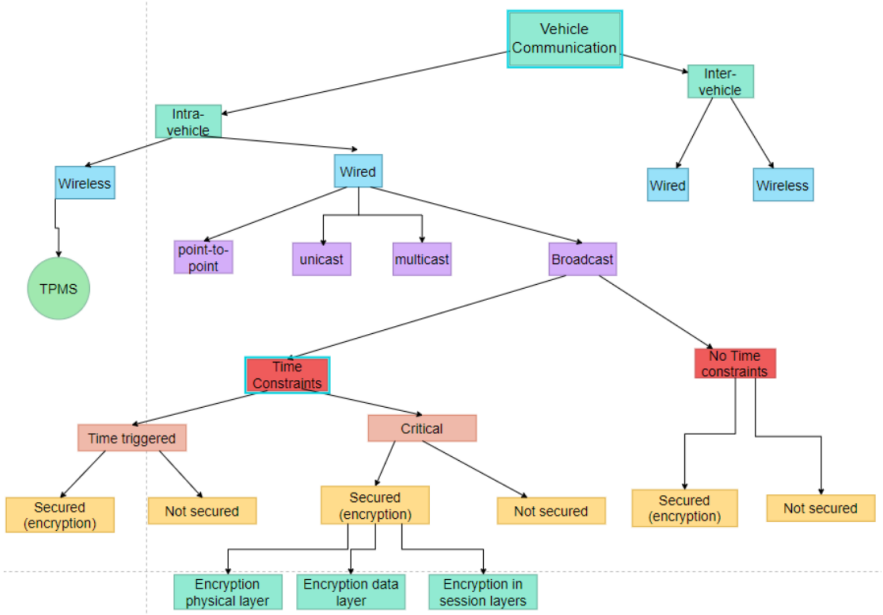

Fig. 5. Pattern Tree 2

further divided up. More specifically it was split up based on the OSI layers. This has been done since encryption can happen on several layers.

While discussing this tree, some more missing elements were found namely, hybrid transmission. The tree contains time triggered communication and event triggered communication. But, also hybrid transmission exists [22, 25–29]. Fur-

thermore, memory and bandwidth constraints were missing too in the tree in figure 5.

We then decided to look at the patterns from another perspective. In a hierarchical way it seems like the higher the pattern decomposition is placed in the tree, the higher its importance is which is not the case.

Furthermore, the tree would become very big and every branch would be a copy of each other. Therefore, we suggested a table instead of a tree. In the table, every decomposition seems of equal importance and it gives a better overview.

The missing element 'hybrid' was added. Also, we made a new decomposition called resource constraints. This decomposition reflects on all the constraints we could find in a system. Namely: timing constraints, memory constraints and bandwidth constraints.

A table 1 was set up. In section 3 every decomposition is discussed in detail.

As mentioned before we would apply recommendations to the patterns. The recommendations have been put in a table displayed in table 2. The recommendations are also discussed in detail in section 3.

Table 1. Pattern 1 decomposition

Location	Intra-vehicle	Inter-vehicle		
Medium	Wired	Wireless		
Data transmission	Point-to-Point	Unicast	Multicast	Broadcast
Resource constraints	Time constraint	Memory constraint	Bandwidth constraint	
Transmission pattern	Event-triggered	Time-triggered	Hybrid	
Data exchange	Service-oriented	Signal-oriented		
Security	Integrity	Confidentiality	Authentication	

Table 2. Pattern Recommendations

Communication protocol
Encryption
Advantages/Disadvantages
Cost
Legacy support
Example

In order to map a pattern on a real use case, we wanted to build an application table. Just as has been done in [36]. We could further sort every use case under a category. But given the time limitations of the report work, it was left as a possible future work.

Alternatively, we could represent the table in another way. We based this solution up on the OSI model. Namely, we can represent the decompositions by every level of the OSI model. Starting from the highest level which would be the application level. Marked in green are the decompositions for the application levels in table 3. The yellow color represents the transport and network layer. The red color represents the physical layer. The only difficulty we faced is the placing of the security decomposition. Security can happen on different layers. An idea would be to split up security protocols in integrity confidentiality and authentication.

Table 3. Patterns Table Add-on

Location	Intra-vehicle	Inter-vehicle		
Data exchange	Service-oriented	Signal-oriented		
Resource constraints	Time constraint	Memory constraint	Bandwidth constraint	
Transmission pattern	Event-triggered	Time-triggered	Hybrid	
Data transmission	Point-to-Point	Unicast	Multicast	Broadcast
Medium	Wired	Wireless		
Security	Integrity	Confidentiality	Authentication	

Table 4. *

Green: application levels
Table 5. *

Yellow: transport and network layer
Table 6. *

Red: physical layer

While reading literature about already existing security protocols, table 7 has been set up. In here the security protocols are placed per OSI level.

Volvo CE provided a domain description for logical reference architecture. In here every domain is displayed and its connections to each other. When reading through the description we took notice that many messages in an architecture would have to travel in between different networks. For example in the logical reference architecture from Volvo CE there is no direct connection between the powertrain domain and the telematics domain. The powertrain logical domain

Table 7. Security sorted per OSI layer

Session Layer	Autosar SecOC / intrusion detection system / firewalls / honeypots
Transport Layer	TLS
Network Layer	IPsec
Data link layer	MACsec / CANAuth Protocol
Physical Layer	CRC / parity / bus guardian

is a name given by Volvo CE to the domain which is used for powertrain related functions with regard to control and actuation. The Telematic domain is a a name given to the domain which is used for wireless connectivity to and from the machine. So in case a powertrain diagnostic message is send to the telematic, it has to hop over another network. Therefore, we added a gateway mechanism to our pattern table. As can be seen in table 8 the gateway mechanism exists out of a homogeneous and a heterogeneous mechanism.

Table 8. Pattern table 2

Location	Intra-vehicle	Inter-vehicle		
Medium	Wired	Wireless		
Gateway Mechanism	Homogeneous	Heterogeneous		
Data transmission	Point-to-Point	Unicast	Multicast	Broadcast
Resource constraints	Time constraint	Memory constraint	Bandwidth constraint	
Transmission pattern	Event-triggered	Time-triggered	Hybrid	
Data exchange	Service-oriented	Signal-oriented		
Security	Integrity	Confidentiality	Authentication	

In addition to the pattern table, the idea was brought up to use feature diagrams. A feature diagram is a visual notation of a feature model. Using feature diagrams can add following advantages. Firstly, it gives a better overview. Secondly, a feature diagram has parental relationships. As displayed in in figure 6, we can have AND and XOR relationships. The engine can only exist out of either a gasoline, electric or hybrid for this particular example. While a comfort function can include all of the under laying examples. Furthermore, the comfort functions are optional. This can be seen on the empty connection dot which connects the 'car' with the 'comfort functions'. If the dot is black it means this part is mandatory. So in case of the car example it is mandatory to have an engine.

According to the previous explained syntax, a feature diagram was build up which is depicted in figure 7. The white squares which can be found at the

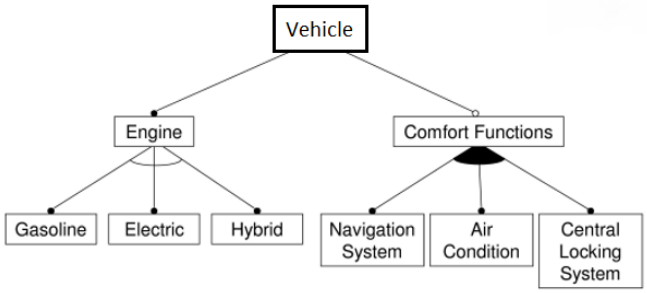

Fig. 6. An Example Feature Diagram

bottom of the connection lines means those entities belong to the same group (e.g., wired and wireless are in the same group which belongs to 'medium').

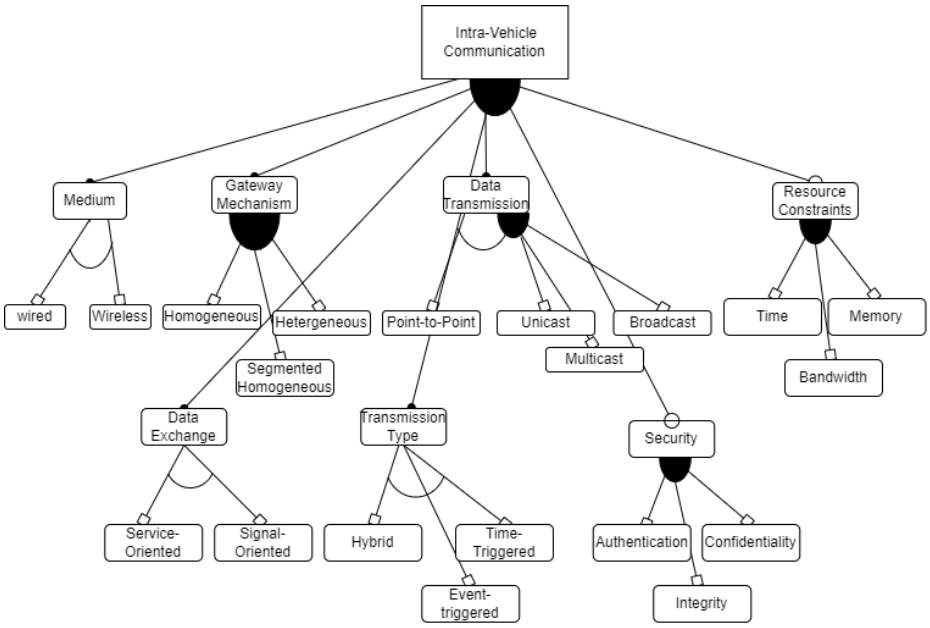

Fig. 7. Feature Diagram

The **medium** is a mandatory entity. All communication will happen either wired or wireless. Therefore, both entities have an 'XOR' relationship with each other.

The **gateway mechanism** is also a mandatory entity. It can be homogeneous or heterogeneous. A message can be send within the network, but in the same time a message can be send to another network. This message travelling to another network will pass a gateway and is therefore called a heterogeneous message. Consequently, the gateway mechanism will have a 'AND' relationship with its entities.

The **data transmission** is also a mandatory entity. There are 2 possible options, namely: it is a point-to-point or something else. This means a point-to-point can not be combined with for example a unicast. Therefore, the entity point-to-point has a 'XOR' relationship with the other entities. On the other hand the other entities have an 'AND' relationship. Which means they can be combined with each other. For example it is possible to have a unicast and a broadcast message in the same network.

The **resource constraints** are not a mandatory entity. Communication might happen without any constraints applied to them. This can be the case in for instance very low data rate communication systems. A typical example here can be the buttons a steering wheel which only transmit a zero or a one when they get triggered. Nevertheless, if constraints are applied they can be grouped in mainly three group entities, which are depicted in figure 7. The group entities have an "AND" relationship with each other because the constraints can exist out of one or more constraints selected from the groups.

The **data exchange** is a mandatory entity. The network can either be service-oriented or signal-oriented. Therefore, the entities have an 'XOR' relationship.

The **security** entity is an additional requirement in a network. Therefore, it is not a mandatory entity. In general security can be divided up in confidentiality, integrity and authentication. Because security may exist out of one or more of the previous described entities, it has an 'AND' relationship.

Every decomposition is discussed in detail in section 3 as well.

Second Iteration After the first analysis, a first attempt was made to map the pattern decompositions on a practical example provided by Volvo CE. This facilitated some new sources for discussions.

A first discussion was held about the difference between a homogeneous and heterogeneous network. This discussion was held because the network in the first use case, which can be found in section 4, could be interpreted either homogeneous or heterogeneous. Moreover, the network consists out of three different CAN busses. Therefore, it can be seen as a heterogeneous network. But if this was the case, than a homogeneous network would rarely occur in an automotive

network which means that it will be less representative and useful within the proposed classification.

For this reason, we came up with the idea of having a segmented homogeneous network in addition to a homogeneous network. A segmented homogeneous network exists out of 'segmented' parts, which means a CAN bus can have for example a domain controller unit in between two different segments of the network. Since the message format stays in CAN format, we do not have a heterogeneous network. Contrary to segmented homogeneous, a heterogeneous network would be obtained if the gateway in between the networks has to converts the message from one protocol format to another format.

A second discussion was held regarding the decomposition 'memory'. Namely, this decomposition could be interpreted in different ways. Firstly, it can be seen as a storage requirement. This would mean that memory reflects to a certain required amount of storage to make the function or controller work. In this case, only an amount of storage needs to be allocated to the memory decomposition.

Secondly, it can be interpreted as a sort of protection system of concurrency control. If a function or controller needs a certain software component to do some calculations with, and another function or controller needs the same software at the same point in time. Than we do not want the software component to become corrupt. In this case, memory would reflect to some kind of concurrency control in where a statement is made which software components are needed in which point in time.

Further discussion facilitated a few more insights regarding the proposed classification. Memory could also reflect on the amount of data produced by for example a high bandwidth sensor (e.g., radars and cameras). Sometimes the processing of this data does not occur directly at the sensor itself. Typically, this happens at a higher-order controller which is connected to the sensor over a network.

This higher-order controller can even be in the cloud. Namely, for machines which do not have such high-order controllers on-board, the data from the sensor can be wirelessly transmitted to the cloud where in higher-order controllers are available. Subsequently, the data would be send back to the machine so it can be used for further calculations (i.e., edge computing). This method would be better applicable on machines which are located in a certain fixed area (e.g., construction equipment). The reason for this is the coverage, if a machine stays in a fixed area the assurance of having full coverage can be calculated. This method would not work (yet) in the automotive domain. Since cars drive around everywhere, even in places where there is no wireless connection, edge computing is not used.

In the first use case, a requirement about redundancy was set by Volvo CE. This led to another discussion about functional safety because this was not discussed before. Therefore, we decided to add function safety to our table of decompositions. A direct remark from Volvo's side was made that Volvo works with the safety standard 13849 instead of the well-known ASIL decompositions

in ISO 26262. More information about the ISO 13849 can be found in section 2. Another remark was made about the transmission type. At first this was called event-triggered or time-triggered, but this caused confusion. Since those concepts come from the real-time embedded systems domain. And the transmission type does not mean any of these concepts. Therefore, the names have changed to respectively sporadic and periodic communication.

The last version of the decomposition table is displayed in table 9.

Table 9. Latest version of the decomposition table

Location	Intra-vehicle	Inter-vehicle		
Medium	Wired	Wireless		
Gateway Mechanism	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

When we started to use the feature diagram from figure 7, a certain issue was found. Namely, it is possible to have wired and wireless communication in a pattern. A solution to address this issue was to create two different feature diagrams. A single hop diagram and a multi hop diagram. Moreover, the single hop diagram depicted in figure 8 would be applied for patterns which do not have a gateway mechanism and therefore are, single hop. The multi hop feature diagram depicted in figure 9 will only differ from the diagram in figure 7 regarding the medium. Namely, the medium in a multi hop pattern might be wired and wireless. While the medium in the single hop feature diagram can only be wired or wireless.

Also, there was no implementation of functional safety in feature diagram 7. So we updated the feature model. The update versions can be found in figure 8 and figure 9. The main difference between the single and multi hop diagram lies in the medium and gateway mechanism. The medium has an XOR relationship in the single hop and an AND relationship in the multi hop. In addition, the gateway mechanism has been removed in the single hop while it is still present in the multi hop.

3.2 Communication Pattern Decomposition

Location The **location** decomposition is based on the physical location where communication will happen. This can either be on-board or off-board. Therefore,

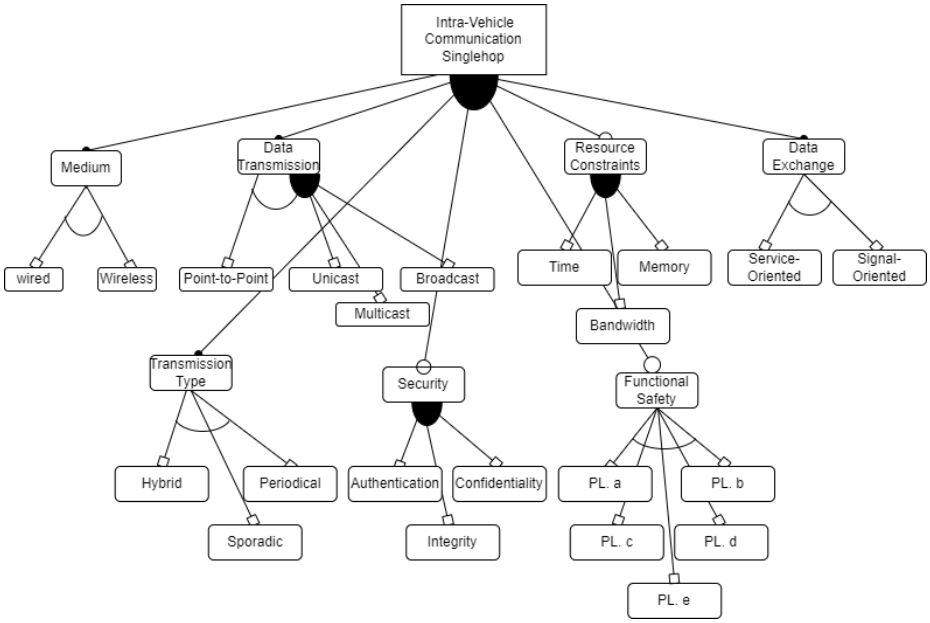

Fig. 8. Single Hop Feature Diagram

the names intra-vehicle and inter-vehicle have been chosen. The intra-vehicle reflects on all the communication that happens on-board (e.g., sensor data, engine management, etc.). While on the other hand inter-vehicle communication reflects on all the communication which happens externally to the vehicle (e.g., remote diagnose data, V2X, cellular connection, etc.).
As explained earlier, the focus of this report will be on the intra-vehicle domain.

Medium The **medium** reflects on the used material wherein communication will happen. Namely, wired or wireless communication. Almost all on-board communication is wired. But there are some wireless applications as well (e.g., tyre pressure monitor system). This report will elaborate more on the wired communication.

Gateway Mechanism The **gateway mechanism** represents the routing of the communication. Moreover, it will display if any gateways or switches are used. Therefore, the name homogeneous and heterogeneous are used. Homogeneous means that communication only happens over the samen network, and no switches or gateways are used to process the communication. Heterogeneous on the other hand, means that communication happens over two or more different networks. A gateway or switch might be present to process the communication. The format of the message will be changed because it is send from one network to another different network. In case a gateway is present but the format of the

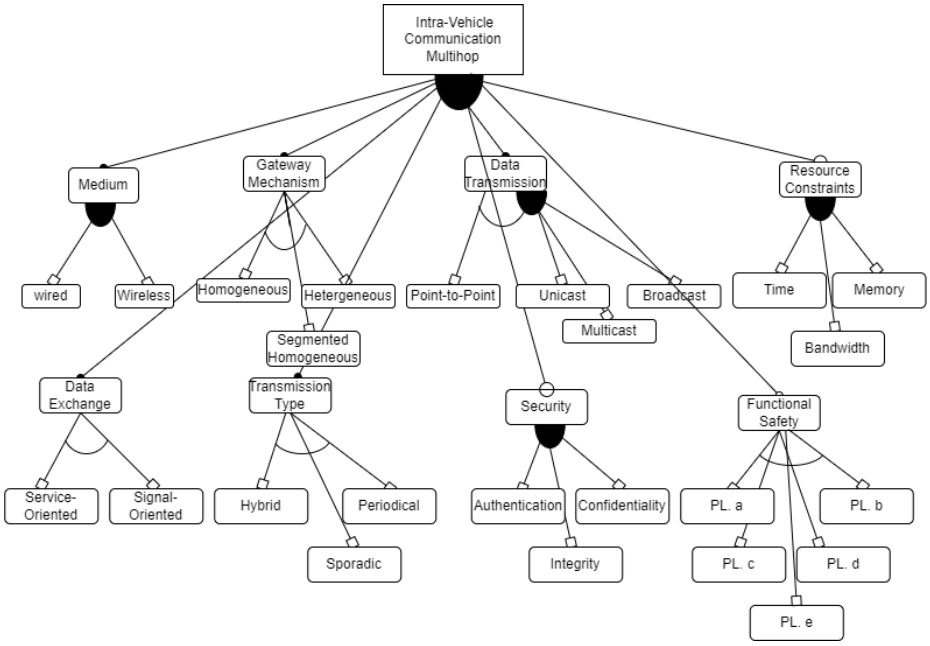

Fig. 9. Multi Hop Feature Diagram

message does not change (e.g., from a CAN network to another CAN network), the name segmented homogeneous will be used.

Data transmission The **data transmission** represents how the communication is set up. Namely, how many receivers will a certain message have. Therefore, we divided this up in 4 sections:

- Point-to-Point.
- Unicast.
- Multicast.
- Broadcast.

A **point-to-point** connection can be seen as a direct connection in between two devices. It is not possible to connect a third device to this direct connection. If, on the other hand a third device is connected to the direct connection it is not seen as point-to-point connection anymore. If communication happens from one device to another in a network with at least three devices it is called a unicast.

A **unicast** definition can be used whenever a message addressed for single recipient in a wider network is applicable. Contrary to unicast, a multicast message may be applicable. A **multicast** message is a message addressed for multiple but not all recipients in a wider network. If a message is send to all recipients in a network it is called a **broadcast** message.

Resource constraints The resource constraints display the possible constraints given by the functional requirements of the system. Three main constraints can be distinguished:

- Bandwidth
- Memory
- Timing

In case the network is subjected to large amounts of data (e.g., radars or cameras), a bandwidth constraint may be put in place. This constraint can regulate the minimum required bandwidth the network need in order to sustain the communication. This will determine which type of communication technology is more suited to implement the functionality.

Another common constraint in the vehicular industry may be a timing constraint. Moreover, many applications require real-time requirements (e.g., brake-by-wire, airbag systems, etc). With the timing constraint we can set a boundary of the maximum latency a certain application or message may have.

The third constraint is memory. Typically, the memory constraint will be used in future architectures. Moreover, the memory constraint defines the amount of data which should be computed outside the vehicle. A vehicle may exist out of a high bandwidth sensor (i.e., radar or camera), and some low processing units. In that case, the data will be processed on a server outside the vehicle (i.e., cloud computing). This would typically be done over a wireless connection such as 5G. Since memory can be considered very broadly, this research wants to focus on this part only. The constraint will not focus on any memory architectures or access methods. As previous explained, this kind of architecture is more future focused. Therefore, it will not be used in today's use cases.

Transmission pattern The **transmission pattern** provides information about the triggering of the communication. There are two main concepts namely: periodic-triggered and sporadic-triggered.

A periodic-triggered signal means the signal is transmitted in a periodical way, which means it is triggered every time a certain period expires. In contrast to periodic-triggered, one can have a sporadic-triggered signal. This means a signal is triggered by a certain external event. Events like braking, accelerating, collision avoidance, etc. are very common to be triggers for sporadic-triggered signals.

Furthermore, a mix of the former and latter may occur. For this instance, the name hybrid has been used. A signal can be simultaneously periodic and sporadic triggered. This is used for example in CANopen, AUTOSAR and HCAN [29].

Data exchange mechanism The **data exchange mechanism** gives an overview on the used mechanism to exchange information. This exists out of service or signal oriented communication. A detailed explanation about service and signal oriented communication can be found in section 2.

Security The **security** gives an overview on the applicable security precautions. A first division is made which represents the Confidentiality, Integrity and Authentication (CIA).

Moreover, confidentiality is to prevent the disclosure of sensitive information from unauthorized people, resources and processes. Integrity reflects on the protection of system information or processes from intentional or accidental modification. Availability reflects on the assurance that the system is accessible by authorized users when needed [7].

Functional Safety The **functional safety** decomposition is based on the safety standard used by Volvo CE. This safety standard is ISO 13849. More detailed information about safety standard ISO 13849 can be found in section 2. As for the use is in this research, the functional safety decomposition will be used in function of the provided requirements. So there will be no fixed concepts available for this decomposition.

3.3 Communication Pattern Recommendations

For every pattern some recommendations will be given. When a system designer will start to create a new system, it can rely on the recommendations given here. Therefore, the designer does not have to start from a completely empty design. A direct consequence will be a speed up of the development process. A table of the recommendations can be seen in table 2.

Communication Protocol Firstly, a recommendation will be given regarding the communication protocol. This recommendation will be given based on the minimum required bandwidth in the pattern. Therefore, a summary of the bitrates per communication protocol has been written out in figure 10.

Protocol	Bitrate	Medium	Protocol
LIN	20Kbps	Single Wire	Serial
CAN	1Mbps	Twisted Pair	CSMA/CR
CAN FD	1-5Mbps	Twisted Pair	CSMA/CR
CAN XL	10Mbps	Twister Pair	CSMA/CR
FlexRay	20 Mbps	Twisted Pair/Optical Fibre	TDMA
MOST	150Mbps	Optical Fibre	TDMA
LVDS	655 Mbps	Twisted Pair	Serial/Parallel
Automotive Ethernet	Up to 1Gbps	Unshieled Twisted Pair	TDMA

Fig. 10. Bitrate Table

Based on the bitrates described in the table, recommendations will be given.

Advantages / Disadvantages Secondly, additional information such as advantages and disadvantages of a certain pattern will be written out in this section. This section might be very broad, but in this research we want to focus us on the very basic advantages and disadvantages such as for instance, easy to maintain or easy to expand sensor network.

Costs Subsequently, a basic indication regarding the cost of the system will be given. This information is based on, for instance, the material which is used (e.g., optical fibre or twisted pair cable). But also the total length of the wires, the amount of ECUs and the development time. This will be represented in a financial aspect.

Encryption In this recommendation we would like to point out which encryption technologies are possible for a certain pattern.

Legacy Support From a architectural point of view, it can be convenient to have a legacy support recommendation. Moreover, the legacy support recommendation gives information about the likelihood that the architecture can work together with an older generation architecture.

Example Finally, the pattern will be represented by some typical examples. These examples will come from communication systems typically found within vehicles. For instance, infotainment systems, pre-crash warning systems, adaptive cruise control and infotainment updates.

4 Industrial Use Cases and Evaluation

This section provides a proof of concept by applying and evaluating the acquired knowledge on industrial use cases. Those use cases are provided by Volvo CE. The feature diagram can be seen as a method to abstract the patterns from the use cases. undoubtedly, the feature diagram will be further refined by every use case which is abstracted.

4.1 Industrial Use Cases

Speed Calculation The first use case provided by Volvo CE was a functionality of the calculation of the machine speed coming from a wheel loader machine. As can be seen in figure 11, there are four domains which will be used. Namely:

- The powertrain is a domain used for powertrain related functions with regard to control and actuation.

– The machine domain is a domain where in the calculations will happen.
– The Human-Machine-Interface (HMI) is a domain used for operator input requests and machine feedback to the operator.
– The telematic is a domain used for wireless connectivity to and from the machine.

Furthermore, there is a redundant CAN connection between the machine, powertrain and HMI domain. This redundant connection connects the primary control unit from every domain. Subsequently, all other ECUs in their domain are connect to their domain primary control unit. As can be seen, the telematic domain is only connected to the HMI primary control unit over a CAN and broadreach Ethernet connection. Therefore, all messages going and coming from the telematic domain will need to go through the HMI primary control unit. Also, there is diagnostic connector (DC) which is connected over CAN to the machine domain and classical Ethernet to the HMI domain.

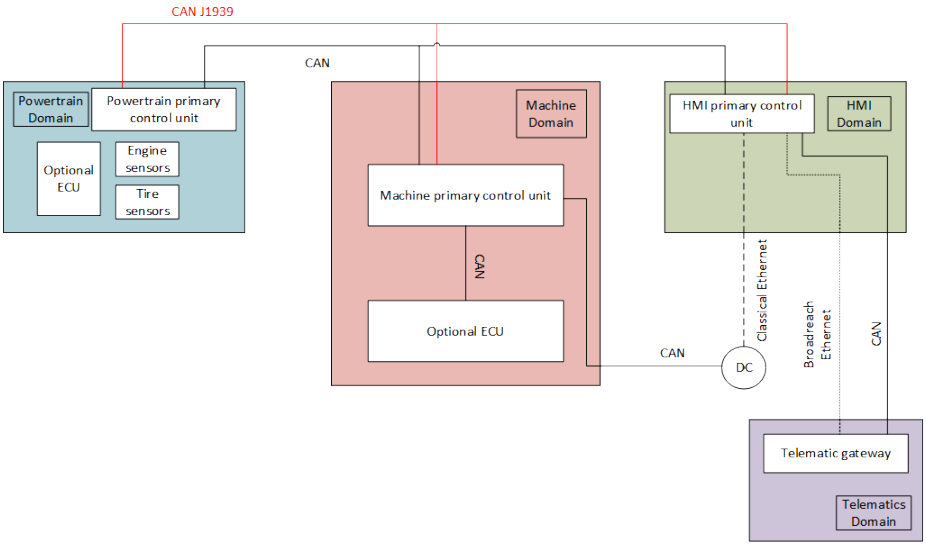

Fig. 11. Speed Calculation Abstraction

An important assumption was made by Volvo, namely: the telematics unit is sufficient secured against any unauthorized access. Therefore, no other security should be considered in this system.

To calculate the machine speed three parameters are necessary. Those three parameters will be transmitted from the powertrain domain to the machine domain. In the machine domain, those parameters will be received by the machine primary control unit. Subsequently, they will be send over a different CAN network to the ECU wherein the speed calculations happen. From this information the decomposition table was filled in which is exhibited in table 10.

Table 10. Machine Speed Calculation Table

Location	Intra-vehicle
Medium	Wired
Gateway Mechanism	Segmented Homogeneous
Data Transmission	Multicast
Resource constraints	Time and Bandwidth
Data exchange	Signal Oriented
Transmission type	Periodic
Security	N/A
Functional Safety	Redundancy

The **location** of the communication is intra-vehicle. The **medium** is wired. For **gateway mechanism** the segmented homogeneous option has been selected. As discussed previously this is not a homogeneous network, and neither is it a heterogeneous network. Moreover, the format of the messages coming from the powertrain domain will not be changed because the message is send to another CAN bus (the machine master ECU will take care of this function). Besides, it is not a homogeneous network because the CAN message is send to another CAN bus. Therefore, the name segmented homogeneous is selected.
The **data transmission** is multicast. Moreover, a CAN network is always multicast.

Volvo CE set a requirement that the machine speed output should be updated every 50ms. For this requirement the time constraint was used.

Furthermore the **data exchange** is signal oriented. The **transmission type** is periodic since all parameters are send periodically. From the assumption made earlier there is no need for security. Volvo CE set a requirement for the redundancy. Namely, the machine speed and machine in motion parameter need redundancy and therefore a redundancy on the CAN network is used.

Two resource constraints can be found, namely a time and a bandwidth constraint. A total of 5 parameters need to be transmitted in order for the machine speed function to function. Therefore, a bandwidth constraint has been applied. Every parameter exists out of 34 bits, which results in a total of 34 * 5 = 170 bits. To calculate the worst-case transmission time the equation found in [29] has been used.

$$C_m = (47 + 8s_m + [\frac{34 + 8s_m - 1}{4}]) \tag{1}$$

Where denotes the time required to transmit a single bit of data on the CAN network. It depends on the speed of the network which has been chosen to be 500kbit/s. This value has been chosen from a industrial point of view. Moreove,

1000kbit/s is rarely used in industry. Therefore the value of will be $1/(500*10^3)$.

The value of Sm will be an integer value starting from 0 to 8. It represents the databytes in a CAN message. So a maximum of 8*8 = 64 bits can be transmitted in 1 CAN frame.

The number 47 comes from the overhead of the protocol, namely the standard CAN identifier frame format. In case it would be an extended CAN frame format the number would be 67 [29].

For this use case a total of 170 bits need to be transmitted in frames of 64 bits each. Which results in 64 + 64 + 42 bits per frame. When transmitting 64 bits in 1 frame the following results were obtained.

Cm = 135 *

Cm = 0.270ms

The worst-case transmission time for the frame with only 42 bits (i.e., 6 bytes), will be:

Cm = 116 *
Cm = 0.232ms

The total worst-case transmission time will be:

2 * 0.270ms + 1 * 0.232ms = 0.772ms

From this result, a conclusion can be made from the designer. In case this number is lower than the desired worst-case transmission time, than the chosen data rate is able to support the transmission of the desired amount of data. Because the worst-case transmission time seemed more relevant than a best-case transmission time, we did not elaborate on a best-case transmission time.

Engine Speed and Vehicle Speed Distribution Log The second use case provided by Volvo CE is a functionality of the engine speed and vehicle speed distribution logging. It is connected to the previous use case in a sense that they use the same domains. Therefore, no new abstraction of the domains is given for this use case. Furthermore, the same assumption will be made about the telematics unit. Another assumption is made that the position data is received from the telematics unit which is connected over an Ethernet and CAN link. The decompositions for this use case are displayed in table 11.

The differences with the previous use case are: The telematic unit is considered to use the Ethernet link instead of the CAN link. This will introduce a message format change which is valuable to consider. Namely, the messages

Table 11. vehicle Speed Distribution Table

Location	Intra-vehicle
Medium	Wired
Gateway Mechanism	Heterogeneous
Data Transmission	Multicast
Resource constraints	Time and Bandwidth
Data exchange	Signal Oriented
Transmission type	Periodic
Security	N/A
Functional Safety	Redundancy

coming from the telematic unit are in an Ethernet form, so they will have to be transformed to a CAN format. Therefore the **gateway mechanism** is heterogeneous.

Furthermore there are still 2 **constraints** namely a bandwidth and a time constraint. The time constraint contains an update of engine speed and machine speed distribution log with a period of 1000ms. The bandwidth constraint is bigger than in the previous use case. Moreover, for this function to operate in normal conditions eleven parameters need to be transmitted.

The equation found in equation 1 will be used to give an approximation of the worst-caste transmission time. A total of 11 * 34 = 374bits will need to be transmitted. This is the equivalent of 47 bytes. In the previous use case the equation resulted in a transmission time of 0.272ms for 8 bytes. Since there are 47 bytes, there will be 5 messages with 8 bytes and 1 message with 7 bytes.

The worst-case transmission time for a CAN message with 7 bytes will be:

Cm = 126 *

Cm = 0.252ms

The total worst-case transmission time will be:

5 * 0.272ms + 1 * 0.252ms = 1.612ms

The transmission type is **periodic**. Analog to the previous example there will be no security and a functional safety requirement namely a redundancy on the communication network.

Engine HMI Presentation and Logging This use cases comes from the same machine as the previous use cases. Also, it uses the same domains. Therefore, the same abstraction will be used for this use case and can be found in figure 11.

Similar to previous use cases, the assumption is made that the telematics unit is considered secure. Therefore, no security against external threats is implemented.

The engine Human-Machine-Interface (HMI) presentation realizes a certain amount of display functions (e.g., engine oil pressure and temperature, engine air filter status and starter motor overheating). These parameters are transmitted from the powertrain domain to the HMI domain. Furthermore, logging data is send from powertrain domain to the machine domain.

An overview of the decompositions is given in table 12.

Table 12. Engine HMI presentation Table

Location	Intra-vehicle
Medium	Wired
Gateway Mechanism	Heterogeneous
Data Transmission	Multicast and Unicast
Resource constraints	Time and Bandwidth
Data exchange	Signal Oriented
Transmission type	Hybrid
Security	N/A
Functional Safety	PL b

The **gateway mechanism** is heterogeneous. Moreover, it is likely that certain sensors in the powertrain domain will be connected to the domain control unit over a different protocol than CAN. Therefore, the DCU will have to change the format of the message to CAN format so it can be send over to the HMI (i.e., heterogeneous).

The **data transmission** is multi and unicast. Because of CAN, multicast messages will be present. Unicast messages might be found in the powertrain domain were low-data sensors transmit their data to the domain controller unit.

Furthermore, a **time and bandwidth constraint** are present. The 'EngOilAndEngAirFiMon' (i.e., engine oil and engine air filter monitoring) needs to be updated every 100ms. A total of 8 parameters will be transmitted over the network possibly causing a bandwidth constraint.

The **transmission type** will be periodic since all parameters are send periodically.

Lastly, the **functional safety** set by Volvo CE is Pl b. More information about the different safety levels can be found in section 2.

4.2 Communication Patterns

Pattern 1 The first pattern is derived from the speed calculation use case which can be found earlier on in this section. The pattern is build out of three

main components. Namely, the sensors, computing platform and an actuator or display. A representation of the pattern is displayed in figure 12.

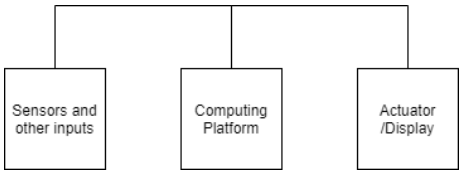

Fig. 12. Pattern 1

The pattern can be seen as a generic pattern for many kind of communications. It belongs to the domain based E/E architectures. This pattern would not be applicable to the distributed architecture, since a distributed platform has its computing platforms distributed in the vehicle and not together in one area. Some recommendations are given in the following table. Note that the legacy support is not elaborated on. The reason for this is because the legacy support is based on its predecessors. One would need more than one patterns to elaborate on the legacy support.

Table 13. Pattern 1 recommendations

Communication protocol	CAN / FlexRay
Encryption	Application level
Advantages/disadvantages	Easy to maintain
Cost	Low
Legacy support	-
Example	Infotainment

Discussion A recommendation is given to use CAN or Flexray for this pattern. But it is up to the designer to decide which protocol should be used. The table found in figure 10 can be a help deciding which protocol is a better fit based on its datarate.

In case CAN is used and there is no real-time constraint present. Once can implement an authentication protocol on the CAN bus. This proposal is called CANAuth [15].

It is also possible to implement a transport layer security (i.e., TLS) over CAN protocol. This would assure an end-to-end security. This proposal has been tested and the outcomes were positive [41].

In addition, encryption can happen on session layer for CAN and FlexRay. Security precautions such as a honeypot and an intrusion detection system can

be implemented. For more detailed information about those precautions, see section 2.

Furthermore, all the computing will be performed in one ECU which is abstracted as computing platform in figure 12. Because of this reason it is possible to implement a fallback strategy. The fallback strategy can be activated from the moment the ECU abstracted as computing platform gets compromised. Moreover, the computing can be moved to other ECUs which have access to the data which should be computed. However, this means that the other ECUs need to have additional available computing power in normal circumstances to be able to deal with the extra required computing power [8].

Pattern 2 A second pattern is derived from the use case engine speed and vehicle speed distribution log which can be found earlier on in this section. The pattern exists out of the same components as pattern number 1, but has an additional component called the additional data. This data is retrieved from an area which is connected over a different communication protocol than 'sensors and other inputs'. And therefore, giving this pattern a heterogeneous gateway mechanism.

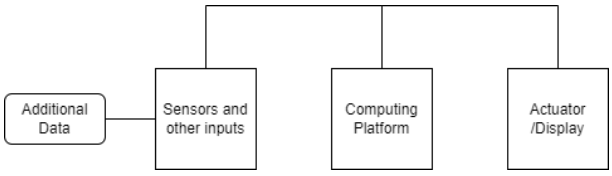

Fig. 13. Pattern 2

This pattern also belongs to the domain centralized E/E architectures. The pattern recommendations are given in following table.

Table 14. Pattern 2 recommendations

Communication protocol	CAN / FlexRay / Automotive Ethernet
Encryption	Application level
Advantages/disadvantages	Scalability
Cost	Medium
Legacy support	-
Example	Infotainment, data exchange in between networks

Discussion Similar to the pattern 1, a fallback strategy can be implemented. Also CANauth and TLS can be implemented in case CAN protocol is used and there are non-real time requirements.

Pattern 3 The third pattern is derived from the use case HMI presentation and logging. The domain control unit acts like a gateway for the low-end sensors connected in the same domain. It is also for this reason this pattern is considered to have a heterogeneous gateway mechanism.

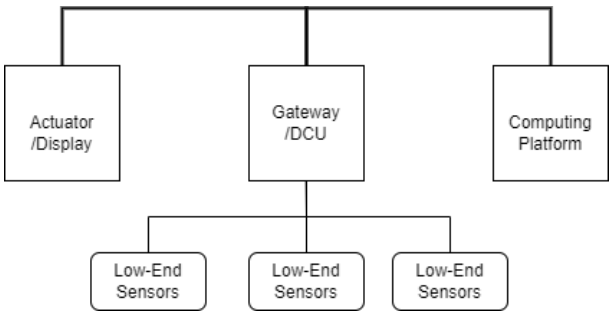

Fig. 14. Pattern 3

Table 15 gives the recommendations for this pattern. Similar to previous patterns, it is up to the architect to decide which protocols suits best. This will mainly be based on the required data throughput.

Table 15. Pattern 3 recommendations

Communication protocol	CAN / FlexRay
Encryption	Application level
Advantages/disadvantages	Easy to maintain / expand sensor network
Cost	Low
Legacy support	-
Example	Sensor network, Engine domain

Discussion In this pattern, a logging takes place. This logged data is saved on a certain memory location. An addition to this system could be a sanity check. This check would contain a procedure which checks the availability of the memory. Moreover, this check could be executed by the startup of the system. If the memory is available, the logging could start. In case the memory is not

available, the logging should not start and a status update could be send to a designated operator.

Another idea is to implement cloud computing. In this case, the computing platform depicted on the right side in figure 14 would disappear. A direct consequence would be a reduction in costs, namely, less ECUs and less wires have to be implemented. Also the complexity of the system in the vehicle would decrease. However this might reduce the complexity of the system in the vehicle, it might increase the complexity of computing the data. Namely, the offloading must work across heterogeneous environments such as different software and hardware architectures [6].

Furthermore, not enough patterns are described to give a detailed recommendation for the decompositions legacy support and cost. The reason for this is because they are relatively to each other. Also, the advantage and disadvantage decomposition could be elaborated on more intensively. But also for this decomposition, it would make more sense to elaborate on it when there are more patterns present.

5 Validation

In this section, threats to the validity and limitations of the performed work are set out.

5.1 Validity

Throughout the research several assumptions were made to limit the scope of the research process. It is important to acknowledge these assumptions and discuss the possible limitations they may introduce to the applicability of the conducted research.

Firstly, only wired intra-vehicle communication systems have been researched (i.e., communication inside the vehicle) . Therefore, some research can still be conducted on the inter-vehicle communication. Even though, the inter-vehicle communication were introduced in the recent years and are therefore better researched than the intra-vehicle communication systems. It is also for this reason we focused our research on the intra-vehicle communication systems.

Secondly, the research was limited to the wired communication systems. We excluded the wireless communication from the work because this was of little interest to Volvo CE. Moreover, the wireless communication inside the vehicle is very limited (i.e., Tyre Monitoring Systems) and would therefore provide us with no additional information regarding communication patterns.

The limitations will not affect the research because they can be seen as side branches of the main research. The research had a extensive focus on the communication inside the vehicle, and therefore the communication outside the vehicle and the wireless communication will not affect the obtained results.

5.2 Threats to Validity

The performed work is based on information acquired from trusted databases (e.g., Springer Link, IEEE Explore, ResearchGate). But also Google Scholar was used. Typically, those objective sources have none to almost none influence from a commercial viewpoint. The methodology used in this research is evaluation. Therefore, the information acquired for this research depend on the integrity of the used sources which are, as previous explained, trusted databases.

Furthermore, the decompositions found in the first iteration in section **??** have been applied to a real life use case provided by Volvo CE. After some comprehensive discussions the table with decompositions was updated to concur better with the use case. The final design of the table (table 9) was chosen to match the requirements provided by Volvo. Even though it was designed to match Volvo's requirements, the results are generalized in a way that is applicable to different communication systems. Moreover, the table is meant to work for all kind of communication systems in different vehicular domains. Since the evaluation has only been done by use cases from Volvo CE the validity for other vehicle domains can not be assured. Similar to the table, the feature diagram 7 has been designed around Volvo CE's requirements but is also applicable to other communication systems.

In this study, names were chosen for the decompositions in the table 9 which are not fixed. Moreover, some of the names were updated by every iteration which was performed and are therefore still prone to changes.

The obtained patterns found in section 4 are designed with focus on generalisation. The patterns are applicable to all kind of vehicle communication systems and are not limited to construction equipment vehicles. Therefore, the patterns will be easy to reproduce. But will also be easy to build upon in any future work focused on communication patterns in E/E architectures.

6 Conclusions

6.1 Research Questions

The research questions asked in section 1 are answered with the accumulated results and evaluations presented throughout this report.

Firstly, the first research question '*How to provide classification for communication patterns used in automotive embedded communication systems?*' has been answered by a consolidation made from several sections in this report. Moreover, a classification is presented throughout this report. Several tables and feature diagrams were made with the aim of classification for communication patterns. Even though, the tables and feature diagrams were designed around communication patterns in construction equipment, they are applicable to other vehicular domains as well.

Secondly, the second research question '*How to allocate functionalities in an E/E architecture with the use of the classification?*' has been answered by giving certain design recommendations. These recommendations were given based on

the previous constructed communication patterns. We set out recommendations which we believed were in the scope of this work. Unfortunately, since the time limit of this work the recommendations were not validated. The boundaries of the recommendations should not be limited to this work only and can be expanded in a future work.

6.2 Conclusion

Over the past decade, communication systems have gained an increasingly prominent role in the vehicular industry. With responsibilities such as functionality, safety and security of such an importance, that it would be impossible to think about vehicles without these communication systems anymore. Unfortunately, there is no shared consensus over a default framework for these kind of communication networks. Therefore, there are opportunities to speed up the design process of the E/E architectures.

In this report, we worked on developing a method for abstracting communication patterns from vehicular use cases. We obtained this method by reading literature about the subject and discussing what could be valuable to us. A first representation was made in the form of a hierarchical tree. Next, a table was designed with different pattern specifications. Subsequently, a feature diagram was made which has the additional benefit of showing internal relations from the decompositions of the pattern. Because of the many discussion which were held, the result of the feature diagram was obtained in an iterative manner.

Then, the method was tested by the means of use cases. Moreover, some use cases were provided by Volvo CE. The feature diagram was applied to these use cases and as a result we obtained some abstracted patterns.

One can work in the other direction as well. A pattern can be taken and further elaborated on. This should result in a decrease of development time, and is therefore the main objective of the patterns. Furthermore, recommendations are given to each pattern with the aim of making system development more efficient. To this end, only one direction was exercised in this work namely abstracting the patterns from an already given architecture.

6.3 Future work

Although this report has presented a comprehensive method for mapping architectural communication patterns, there is a clear possibility for future development. The research conducted throughout this work has been limited by several assumptions. In future works, the assumptions can be removed to get a broader spectrum of communication patterns. Firstly, the assumption of no inter-vehicle communication can be removed. Secondly, the assumption of only wired communication can be lifted.

One could introduce the aspect of cloud/edge computing if the assumption of inter-vehicle is relaxed. This can be very beneficial for architects since these are concepts which will be elaborated on in the (nearby) future.

In the pattern recommendations table, recommendations in the form of advantages and disadvantages are given. Since this work only includes the elaboration of three use cases, not many advantages and disadvantages could be given. Moreover, the advantages and disadvantages are relatively to other communication patterns. Therefore, these will gain value once more use cases and patterns are elaborated on. We included this in the research for future purposes.

Analog to the advantages and disadvantages, the legacy support and costs are also relatively to each other. Therefore, one could give a better approximation of those decompositions after more use cases and patterns are elaborated on.

Once one obtained more patterns, it could be beneficial to create an application table. This has been done in [36]. Every pattern could be placed in a certain category, for instance, active safety and infotainment.

Also, when having enough patterns, a pattern catalogue can be made. Similar to design patterns in software. This pattern catalogue will contain all patterns, split up in categories. In addition to the pattern catalogue, a decision tree can be used. This decision tree would be a guidance for architects in finding the right pattern.

Furthermore, the communication patterns which are described in this work should not be limited to only construction equipment. The patterns are build up in such a way that they are applicable to other vehicular systems as well. Vehicles like cars, motorcycles, buses and trucks are very common. Even though the patterns are derived from a vehicle on the ground, they might as well be applicable to a broader spectrum such as aircraft and spacecraft vehicles. Also, boats and railed vehicles can benefit from this. In a future work, one could make a comparison wherein a mapping is made between the patterns and different kind of vehicles. Also validation can be done between these cross-domain patterns.

Lastly, this work did not elaborate on functional safety. However, we included functional safety levels in the latest version of the decomposition table (table 9). There is no further elaboration on this subject. In future works, functional safety could be implemented in the patterns.

References

1. Robert Bosch GmbH, CAN with Flexible Data-Rate (CAN FD), White Paper, Ver. 1.1., 2011
2. Autosar technical overview version 4.3, 2016, http://autosar.org, accessed: 2021-12-29
3. Controller area network extra long (can xl), http://canopen.org/cn/can-knowledge/can/can-xl, accessed: 2021-12-29
4. Guest editorial embedded and networked systems for intelligent vehicles and robots. IEEE Transactions on Industrial Informatics 15(2), 1035–1037 (2019)
5. Ashjaei, M., Bello, L., Daneshtalab, M., Patti, G., Saponara, S., Mubeen, S.: Time-sensitive networking in automotive embedded systems: State of the art and research opportunities. Journal of Systems Architecture 117, 102137 (2021), https://www.sciencedirect.com/science/article/pii/S1383762121001028

6. Ashok, A., Steenkiste, P., Bai, F.: Enabling vehicular applications using cloud services through adaptive computation offloading. In: the 6th International Workshop. pp. 1–7 (September 2015)
7. Avizienis, A., Laprie, J.C., Randell, B., Landwehr, C.: Basic concepts and taxonomy of dependable and secure computing. IEEE Transactions on Dependable and Secure Computing 1(1), 11–33 (2004)
8. Bandur, V., Pantelic, V., Dawson, M., Schaap, A., Wasacz, B., Lawford, M.: A domain-centralized automotive powertrain e/e architecture. In: SAE WCX Digital Summit (April 2021)
9. Bello, L.L., Mariani, R., Mubeen, S., Saponara, S.: Recent advances and trends in on-board embedded and networked automotive systems. IEEE Transactions on Industrial Informatics 15(2), 1038–1051 (February 2019)
10. Bucaioni, A., Pelliccione, P.: Technical architectures for automotive systems. In: 2020 IEEE International Conference on Software Architecture (ICSA). pp. 46–57 (2020)
11. Cooperation, M.: MOST Specifications . Bercker, Karlsruhe Germany (2011)
12. Garnatz, O., Decker, P.: Can xl provides the basis for seamless cooperation with ethernet. Vector (May 2020)
13. Garnatz, O., Decker, P.: Ip concepts with can xl. Vector (2020)
14. Gopu, G.L., Kavitha, K.V., Joy, J.: Service oriented architecture based connectivity of automotive ecus. In: 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT). pp. 1–4 (2016)
15. Herrewege, A., Singelée, D., Verbauwhede, I.: Canauth - a simple, backward compatible broadcast authentication protocol for can bus. In: ECRYPT Workshop on Lightweight Cryptography 2011. p. 7 (January 2011)
16. Huq, S.B.: An overview of lvds technology. National Semiconductor Application Note (1998)
17. ISO 11898-1: Road Vehicles ? interchange of digital information ? controller area network (CAN) for high-speed communication, ISO Standard-11898, Nov. 1993.
18. ISO 17458-1: Road vehicles — FlexRay communications system, ISO Standard-17458, Feb. 2013.
19. Kleberger, P., Olovsson, T., Jonsson, E.: Security aspects of the in-vehicle network in the connected car. In: 2011 IEEE Intelligent Vehicles Symposium (IV). pp. 528–533 (2011)
20. Leen, G., Heffernan, D.: Time-triggered controller area network. Computing Control Engineering Journal 12, 245 – 256 (01 2002)
21. Mody, M., Jones, J., Chitnis, K., Sagar, R., Shurtz, G., Dutt, Y., Koul, M., Biju, M., Dubey, A.: Understanding vehicle e/e architecture topologies for automated driving: System partitioning and tradeoff parameters. Electronic Imaging 2018, 1–5 (01 2018)
22. Mubeen, S., Jukka, M.T., Mikael, S.: Extending schedulability analysis of controller area network (can) for mixed (periodic/sporadic) messages. In: ETFA2011. pp. 1–10 (2011)
23. Mubeen, S., Lisova, E., Feljan, A.V.: A perspective on ensuring predictability in time-critical and secure cooperative cyber physical systems. In: 2019 IEEE International Conference on Industrial Technology (ICIT). pp. 1379–1384 (2019)
24. Mubeen, S., Lisova, E., Vulgarakis Feljan, A.: Timing predictability and security in safety-critical industrial cyber-physical systems: A position paper. Applied Sciences 10(9) (2020)

25. Mubeen, S., Mäki-Turja, J., Sjödin, M.: Extending response-time analysis of controller area network (can) with fifo queues for mixed messages. In: the 16th IEEE Conference on Emerging Technologies and Factory Automation (ETFA), 2011, WIP. pp. 1–4. IEEE (September 2011)
26. Mubeen, S., Mäki-Turja, J., Sjödin, M.: Response-time analysis of mixed messages in controller area network with priority- and fifo-queued nodes. In: 2012 9th IEEE International Workshop on Factory Communication Systems (WFCS). pp. 23–32. IEEE (May 2012)
27. Mubeen, S., Mäki-Turja, J., Sjödin, M.: Extending worst case response-time analysis for mixed messages in controller area network with priority and fifo queues. IEEE Access 2, 365–380 (2014)
28. Mubeen, S., Mäki-Turja, J., Sjödin, M.: Mps-can analyzer: Integrated implementation of response-time analyses for controller area network. Journal of Systems Architecture 60(10), 828–841 (2014), https://www.sciencedirect.com/science/article/pii/S138376211400068X
29. Mubeen, S., Mäki-Turja, J., Sjödin, M.: Integrating mixed transmission and practical limitations with the worst-case response-time analysis for controller area network. Journal of Systems and Software 99, 66–84 (2015), https://www.sciencedirect.com/science/article/pii/S0164121214001952
30. Oka, D., Larson, U., Picasso, F., Jonsson, E.: A first simulation of attacks in the automotive network communications protocol flexray. In: Proceedings of the International Workshop on Computational Intelligence in Security for Information Systems. pp. 84–91 (January 2008)
31. Porras-Vázquez, A., Romero-Pérez, J.A.: 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 174, 60–70 (June 2018), https://www.sciencedirect.com/science/article/pii/S0951832017308359
32. Robert Bosch GmbH: CAN with Flexible Data-Rate (CAN FD), White Paper, Ver. 1.1. (2011)
33. Ruff, M.: Evolution of local interconnect network (lin) solutions. In: 2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No.03CH37484). vol. 5, pp. 3382–3389 Vol.5 (2003)
34. Rumez, M., Grimm, D., Kriesten, R., Sax, E.: An overview of automotive service-oriented architectures and implications for security countermeasures. IEEE Access 8, 221852–221870 (2020)
35. Rylander, A., Wallin, E.: Lin – local interconnect network – for use as sub-bus in volvo trucks. CHALMERS UNIVERSITY OF TECHNOLOGY (2003)
36. Schoch, E., Kargl, F., Weber, M., Leinmüller, T.: Communication patterns in vanets. Communications Magazine, IEEE 46, 119 – 125 (December 2008)
37. Tuohy, S., Glavin, M., Hughes, C., Jones, E., Trivedi, M., Kilmartin, L.: Intra-vehicle networks: A review. IEEE Transactions on Intelligent Transportation Systems 16(2), 534–545 (2015)
38. Tuohy, S., Glavin, M., Jones, E., Trivedi, M., Kilmartin, L.: Next generation wired intra-vehicle networks, a review. In: 2013 IEEE Intelligent Vehicles Symposium (IV). pp. 777–782 (2013)
39. Verendel, V., Nilsson, D.K., Larson, U.E., Jonsson, E.: An approach to using honeypots in in-vehicle networks. In: 2008 IEEE 68th Vehicular Technology Conference. pp. 1–5 (2008)
40. Wendt, T., Bernhart, W., Behl, J., Mishoulam, D., Goldsmith, E.: consolidation in vehicle electronic architectures (July 2015)

41. Yushev, A., Barghash, M., Nguyen, M.P., Walz, A., Sikora, A.: Tls-over-can: An experimental study of internet-grade end-to-end communication security for can networks. IFAC-PapersOnLine 51(6), 96–101 (2018), https://www.sciencedirect.com/science/article/pii/S2405896318308802, 15th IFAC Conference on Programmable Devices and Embedded Systems PDeS 2018