Beyond connected cars: A systems of systems perspective

Science of Computer Programming 191 (2020) 102414

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

Beyond connected cars: A systems of systems perspective

Patrizio Pelliccione [a,b,*], Eric Knauss [b], S. Magnus Ågren [b], Rogardt Heldal [b,c], Carl Bergenhem [d], Alexey Vinel [e,c], Oliver Brunnegård [f]

[a] Università degli Studi dell'Aquila, DISIM, Italy
[b] Chalmers | Gothenburg University Gothenburg, Sweden
[c] Western Norway University of Applied Sciences, Bergen, Norway
[d] Qamcom Research & Technology AB, Gothenburg, Sweden
[e] Halmstad University, Halmstad, Sweden
[f] Veoneer Sweden AB, Vårgårda, Sweden

ARTICLE INFO

Article history:
Received 22 March 2019
Received in revised form 9 January 2020
Accepted 23 January 2020
Available online 3 February 2020

Keywords:
Software engineering
Systems of systems
Automotive
Architecture framework
Software architecture

ABSTRACT

The automotive domain is rapidly changing in the last years. Among the different challenges OEMs (i.e. the vehicle manufacturers) are facing, vehicles are evolving into systems of systems. In fact, over the last years vehicles have evolved from disconnected and "blind" systems to systems that are (i) able to sense the surrounding environment and (ii) connected with other vehicles, the city, pedestrians, cyclists, etc. Future transportation systems can be seen as a System of Systems (SoS). In an SoS, constituent systems, i.e. the units that compose an SoS, can act as standalone systems, but their cooperation enables new emerging and promising scenarios. While this trend creates new opportunities, it also poses a risk to compromise key qualities such as safety, security, and privacy.

In this paper we focus on the automotive domain and we investigate how to engineer and architect cars in order to build them as constituents of future transportation systems. Our contribution is an architectural viewpoint for System of Systems, which we demonstrate based on an automotive example. Moreover, we contribute a functional reference architecture for cars as constituents of an SoS. This reference architecture can be considered as an imprinting for the implementations that would be devised in specific projects and contexts. We also point out the necessity for a collaboration among different OEMs and with other relevant stakeholders, such as road authorities and smart cities, to properly engineer systems of systems composed of cars, trucks, roads, pedestrians, etc. This work is realized in the context of two Swedish projects coordinated by Volvo Cars and involving some universities and research centers in Sweden and many suppliers of the OEM, including Autoliv, Arccore, Combitech, Cybercom, Knowit, Prevas, ÅF-Technology, Semcom, and Qamcom.

1. Introduction

A System of Systems (SoS) is a collection of often pre-existing and/or independently owned and managed systems that collectively offer a service that is based on their collaboration [22,41]. Prominent examples of SoSs include intelligent trans-

* Corresponding author.
E-mail addresses: patrizio.pelliccione@univaq.it, patrizio.pelliccione@gu.se (P. Pelliccione), eric.knauss@gu.se (E. Knauss), magnus.agren@chalmers.se (S.M. Ågren), heldal@chalmers.se, rogardt.heldal@hvl.no (R. Heldal), carl.bergenhem@qamcom.se (C. Bergenhem), alexey.vinel@hvl.no, alexey.vinel@hh.se (A. Vinel), oliver.brunnegard@veoneer.com (O. Brunnegård).

https://doi.org/10.1016/j.scico.2020.102414
0167-6423/

port systems, integrated air defense networks, applications in healthcare, and emergency services. The units that compose an SoS are systems themselves and are called constituents. SoSs may be formed and evolve as triggered by changes in their operating environment and/or in the goals of the autonomous constituent systems [22,10]. The overall SoS evolution might affect the structure and composition of the constituents, functionalities offered, and/or the functionalities quality. Collaboration between SoSs enables new capabilities, but potential interdependencies imply sensitivity to the correctness of the information given to other systems, and that failures can cascade throughout the SoS, creating additional system failures or development delays.

Future transportation systems will be a heterogeneous mix of (smart) systems with varying connectivity and interoperability. A mix of new technologies and legacy systems will co-exist and thus realize a variety of scenarios involving not only connected cars but also road infrastructures, pedestrians, cyclists, etc. In other words, future transportation systems can be seen as SoS [22,41,10], where each constituent system – one of the units that compose an SoS – can act as a standalone system, but the cooperation among the constituent systems enables new emerging and promising scenarios. A constituent system can also be part of several different SoS concurrently, each SoS with its own functional purpose, and with different priority and timing constraints depending, e.g., on whether its functional purpose is operational, tactical or strategical.

A constituent system within an SoS has a value in itself, has its own goals, can be independently managed, and can be used outside the SoS context. Constituents potentially need to sacrifice their ability to operate independently in order to assure the satisfaction of a centrally managed and agreed SoS goal. Moreover, often an SoS is owned and evolved by different organizations and constituents of an SoS are, more often than not, at different points in their life cycles. Collaboration among constituent systems of an SoS enables new capabilities, but interdependency requires (i) unambiguous interpretation of shared data, (ii) agreement on how to use data to achieve the agreed SoS goal, (iii) techniques to avoid the cascading of failures throughout the SoS, etc.

There are two distinct perspectives from which to consider SoS in the automotive domain:

- The perspective of the car as a constituent of the SoS, which aims at giving an answer to the following question: *How to engineer a car so as to be part of a system of systems?*
- The perspective of the overall SoS, which aims at giving an answer to the following question: *How to engineer the SoS so that the collaboration among the various constituent systems will achieve the SoS goals?*

This paper focuses on the first perspective: the car as a constituent of the SoS. It provides a bottom-up analysis of technologies that are likely needed in future Systems of Systems for cars. The bottom-up analysis emphasizes the importance of focusing on a single system within a SoS in order to identify the role of each constituent system, how it should be engineered in order to be part of the SoS, the ownership of the data shared within the SoS, privacy, and security concerns. Further, automotive systems are characterized by a large percentage of legacy which is also expected to be a major part of future cars. One can expect that most of the development in the upcoming decade will be mainly iterative, even if disruptive technologies, services, and business models may appear in the same timeframe. Given that the majority of the assumptions in this paper are relevant we can foresee a number of requirements for the vehicles and the infra-structure meaning both ICT infrastructure and the physical infrastructure such as roads, signs, buildings, etc.

Our contribution is realized as an architectural viewpoint, which is integrated with the architecture framework proposed in [45] and defined according to the standard ISO/IEC/IEEE 42010 [28]. We start by presenting some motivating scenarios, then we present concerns and stakeholders, and we define model kinds. In order to define the model kinds, we first identify the concepts that need to be covered by the model kinds and then we discuss the tools and notations that can be used to define architecture models. We conclude the description of the viewpoint with the presentation of the correspondence rules among this viewpoint and other potential viewpoints. We also present a demonstrator that is used to demonstrate the value of the viewpoint. To give more concreteness to the paper, we also present a functional reference architecture for a vehicle that plays a role as constituent system of an SoS. This reference architecture has been defined taking into account the concepts that we elicited and presented as elements that need to be covered in the model kinds. This reference architecture can be considered as a blueprint for defining architecture solutions that will be defined according to specific constraints of a particular context and OEM.[1] The reference architecture focuses on a specific aspect since it is a functional architecture, and, therefore, it shows the logical decomposition of the system into components and sub-components, as well as the data-flows between them.

The second viewpoint is briefly discussed in Section 8. This implies the opportunities and the steps that should be performed in order to address this viewpoint, i.e. on how to engineer the overall SoS.

This work is carried out in the context of two Swedish projects, *Next-Generation Electrical Architecture (NGEA)* and *NGEA step2*. These projects are coordinated by Volvo Cars and involve some universities and research centers in Sweden and many suppliers in the automotive domain, including Autoliv, Arccore, Combitech, Cybercom, Knowit, Prevas, ÅF-Technology, Semcom, and Qamcom. The main objectives of the projects are to investigate:

[1] In the automotive domain, OEM refers to the manufacturer of the original equipment, that is, the party that assembles and integrates parts from suppliers, during the construction of a new vehicle. Colloquially, the OEM is the “car manufacturer”.

1. the transition of Volvo Cars towards continuous integration and deployment;
2. new business models and innovative ways of working within the automotive ecosystem;
3. vehicles as part of a System of Systems.

This work reports about the knowledge acquired during frequent meetings of the project,[2] focus groups involving both researchers and engineers from Volvo and suppliers, interviews with selected engineers, and study of relevant literature.

This paper is an extension of [46], which presents a preliminary version of the SoS viewpoint. In this paper we extended and consolidated the overall SoS viewpoint thanks to the additional knowledge we acquired through our collaboration with Volvo cars and many other companies. We also added a demonstrator that implements a challenging SoS scenario with the use of real vehicles and that put in practice our SoS viewpoint. We also added a section providing the state of the art in SoS in the automotive domain. Moreover, we contribute a functional reference architecture for vehicles that need to be integrated in a SoS. Finally, we added the perspective of engineering the SoS (see section 8), while the rest of the paper focuses on how to architect a vehicle so to be a constituent system of an SoS. The proposed architecture viewpoint is integrated with the architecture framework proposed in [45].

Paper structure: The paper is structured as follows. Section 2 presents the challenges the automotive domain is facing today. Section 3 provides a state of the art in SoS in the automotive domain. Section 4 highlights the importance of the electrical and software architecture to deal with the challenges highlighted in Section 2. Moreover, this section describes the architecture framework we are building together with Volvo cars. Section 5 focuses on the SoS viewpoint and, as mentioned before, on the main building blocks to enable the car to be part of an SoS. Section 6 presents a demonstrator that puts the SoS in practice in a challenging scenario that makes use of real vehicles. Section 7 shows the functional reference architecture for cars that are part of an SoS. Section 8 focuses on the perspective and challenges on how to engineer an SoS. Finally, Section 9 discusses concluding remarks and highlights future research directions.

2. Challenges the automotive domain

Many emergent business and technological needs are today shocking the automotive domain. Traditional OEMs are realizing that they should increasingly invest in software since 80% to 90% of the innovation within the automotive industry is driven by electronics and software [61]. The development of modern cars has to cope with a large number of concerns, including safety, security, variability management, electrification, autonomy, networking, costs, weight, etc. Moreover, an increasing number of people are involved in software development projects and this imposes additional challenges since the development and design can no longer be controlled, or even understood, in detail by a single group. In the following we discuss some of the main challenges of the automotive domain.

- **Autonomy** – Systems related to the automation of driving aim to provide the human driver with different levels of support in the driving of the vehicle. SAE J3016 distinguishes six levels of vehicle automation [50]. At level 0 a human driver is entirely responsible for the driving task, e.g. steering, throttle, and braking, and this level is referred as no automation. At level 1, known as driver assistance, the vehicle can perform some control function but not everywhere. At level 2, partial automation, the vehicle can handle specified parts of the driving task but the driver is expected to monitor the system and take over in case of faults. In these first three levels, the human driver monitors the environment. In the next three levels, the monitoring of the environment is under the control of the Autonomous Driving System (ADS). At level 3, conditional automation, the ADS monitors the surroundings, but can a short notice hand over the driving task to the human driver. This implies that the driver needs to continually monitor the system and be prepared to take over control of the vehicle (i.e. manual control) at a short notice. At level 4, known as high automation, the vehicle is fully autonomous but only in limited Operational Design Domain (ODD). An example is a function that can handle any situation on specified high-ways; but does not handle entering or leaving the high-way. The ODD specification is not limited to geography, but also includes e.g. speed, scenario, environment, etc. For instance, while on the high-way, any situation with the ODD will be safely handled by the ADS. System failures and e.g. deteriorating conditions that could surpass the limits of the ODD are handled by the ADS e.g. with fall-back maneuver that leads to a safe state. This implies that the driver does not need to continually monitor the system, i.e. she/he can do other tasks. Finally, the full automation is reached at level 5; the ADS has an unlimited ODD and is responsible for the entire trip. An example is an ADS that can handle any trip from start to destination.
  To date, almost all of the vehicles in circulation settle on the levels comprised between level 0 and 2 (level 2 is defined as “partial automation”), in which the systems are limited to assist the driver without replacing him. The path towards fully autonomous vehicles is marked by incremental advances in different categories of assistance. Reaching a level 5 of autonomy is more an evolutionary path rather than a revolution from one day to another. An example of strategy is incrementally expanding the ODD to gradually cover more conditions, area, etc. OEMs are establishing partnerships with suppliers focusing on autonomous vehicles. For example, Zenuity is a joint venture on self-driving cars between

[2] To give an estimate, on average we met our industrial partners around 8 hours a week, 40 weeks per year, during the 4 years of the project.

Volvo Cars and the supplier Autoliv,[3] BMW announced the alliance with Intel and MobileEye with the plan to bring self-driving cars on the road by 2021.[4] Mercedes joins the forces with Bosch, one of the biggest automotive suppliers, to develop level 4 and level 5 vehicles.[5] Google and Tesla are among the pioneers to push the autonomous driving-related software into the market but at least 46 corporations are working on autonomous vehicles.[6] Recently even Amazon is investing in autonomous cars.[7] Most of them had promised to put the autonomous car on the road by 2021, assuming that the necessary regulations will be put in place by then.

The Swedish national-funded research project FUSE [30] (FUnctional Safety and Evolvable architectures for autonomy) was run between 2014 and 2016. The motivating reason for the project was that tomorrow's vehicles should be capable to drive autonomously and that there are a number of questions to solve before this can happen in a safe and efficient manner. The FUSE project, in particular, focuses on architectures and functional safety for autonomous driving systems. The functional architecture proposed in FUSE is described in [4] and is elaborated in [9]. The relation between autonomy and architecture, specifically why evolution of the architecture is needed to support autonomous functions, is commented in [13]

- **Safety** – Safety is one of the major concerns within the automotive domain globally. It has been shown by WHO (World Health Organization) that road traffic injuries are the top cause for death among people 15-29 years of age [67]. When looking at vehicles, the number of sophisticated safety systems is increasing and this has demonstrated a clear safety benefit. However, it is also known that almost half of all fatalities can affect the so-called Vulnerable Road Users (VRU), e.g. pedestrians, cyclists, and powered two-wheelers. Traditionally safety is focusing on vehicles, and consequently, VRUs do not get the same benefit of technology since they generally are not equipped with safety restraints and active safety systems. VRUs are becoming part of future transportation systems through sensors in infrastructure systems but also through connected devices (Internet-of-things).
- **Dependability** – A car being able to act as a member of an SoS will enable many different kinds of services directed to (i) cars, (ii) their drivers and (iii) other end-users of future transportation systems, e.g. the VRU. Many of these services must be dependable to ensure safety. It is important to distinguish between the different levels of Quality of Service (QoS) needs and especially safety requirements that these services give rise to. In general, safety-critical services put very high demands on security, trust, and dependability properties in general. Much of the fulfillment of these properties in its turn depends on sufficient QoS of the connectivity between the SoS constituents involved.
- **Comfort** – Automation of driving became a big buzz in recent years. While improved safety is given as an argument for automation, the main motivations for it are comfort and energy efficiency. As an example, 30% of the traffic in congested areas can be vehicles that are just looking for parking spots [54]. This is also considered a burden (i.e. an unwanted task) for many drivers. Moreover, the view of the car as a status symbol is changing and, in some parts of the world, young people no longer consider having a car to be an important value. This trend is attested by many new actors on the market that provide mobility as a service.
- **Efficiency** – One aspect of efficiency is that of efficient traffic flow. Traffic jams and congestion have been a growing problem in the world, and the road infrastructures are not able to keep up with the rapid increase of demands in major parts of the world due to various reasons such as space, environment, and cost. Electrification of vehicle power-trains is an increasing trend [25], which related to energy efficiency – as electrified power-trains are generally more fuel-efficient than with combustion engines. Charging infrastructure is a new challenge and such infrastructure should be linked to a system to e.g. avoid long waiting times at charging stations.
- **Human Interaction** – When vehicles are automated, a challenge is to make them socially acceptable. In [64], Vinkhuyzen describes a case study of interaction between human road users and automated vehicles. How to interact with pedestrians and other VRU is subject to a social code and can even be different within countries. The same goes for negotiations between vehicles, for example merging (zipper) in one lane before roadworks is subject to cultural aspects. Hence, a System of Systems, consisting of e.g. automated vehicles, needs to adapt to the evolving needs of future transportation system users in order to be socially acceptable.

3. State of the art in systems of systems in the automotive domain

The U.S. Department of Defense categorizes SoSs in four categories, according to the degree of managerial control. This determines how adaptable and cooperative each constituent system will be with respect to requirements, interfaces, data formats, and technologies of the SoS. The categories are:

[3] https://goo.gl/ecuG9G.
[4] https://goo.gl/Fje4P2.
[5] https://goo.gl/EChy4s.
[6] https://goo.gl/AgUgWH.
[7] https://goo.gl/tFvRhS.

- **Virtual SoSs** lack a central management authority and a consensus about the objective.
- **Collaborative SoSs** are characterized by constituents that interact more or less voluntarily to fulfill a shared objective.
- **Acknowledged SoSs** have established consensus about objectives, a designated manager, and dedicated resources for the development, management, and/or governance of the SoS. Constituents retain their independent ownership, objectives, funding, and development approaches. Changes in the constituents are collaboratively coordinated and agreed with the entire SoS.
- **Directed SoSs** are characterized by an integrated SoS, which is built and centrally managed to fulfill objectives for which they have established consensus. It is centrally managed to continue to fulfill those purposes as well as any new ones the system owners wish to address. Constituents maintain an ability to operate independently, but their normal operational mode is subordinated to the centrally managed objective.

The categories "Virtual", "Collaborative", and "Directed" are originally defined in [10], while the "Acknowledged" type has been proposed in [38]. As mentioned before, a categorization is required in order to understand how to build an SoS and to guide the selection of architecting principles [15].

System-of-Systems Engineering (SoSE) [15] is an emerging discipline of the last years that addresses the development, operation, and maintenance of SoSs. SoSE is a specialization of systems engineering and must balance SoS needs with individual system needs. The community around SoSE is active as testified by international events and activities.[8] Moreover, the European Commission has contributed to the development of the field by supporting several projects.[9]

To mention some examples of EU projects, COMPASS and DANSE focus on model-based methods for modeling the architecture and functionality of SoSs, T-AREA-SoS is an agenda-setting project on transatlantic cooperation, ROAD2SOS and CPSOS aim at defining a roadmap on research and innovation in the field, AMADEOS aims at defining an architecture for evolutionary open SoS, LOCAL4GLOBAL aims at operating on constituents to optimize globally the SoS, DYMASOS aims at dynamically managing physically coupled SoS. Nevertheless, although lively, SoSE is a young discipline, implying that general lessons and patterns that cut across applications remain to be learned; as stated in [15]. Here, a structured view of the state of the art in model-based techniques in SoS engineering is given, and identifies challenges for research in this field. A knowledge map on the development of the system of systems research field is given by You in [72]. Another analysis of the developments in the field of SoS (between 1926-2011) is given by Jaradat in [29].

3.1. Automotive domain

Within the automotive domain there are few examples of systems of systems, as summarized below:

- Embedded Automotive Systems – the car itself is a system of systems, i.e. cars as a collection of independent embedded systems on wheels [52]. This is investigated within the EU Verdi project – http://www.verdi-fp7.eu.
- Intelligent Transport Systems (ITS) as systems of systems, e.g.:
  - Transport integrated platform to manage different kinds of subsystems (each module of the platform) and different types of entities, such as electronic devices, and even drivers [6].
  - Framework for Future Integrated Transport System Architecture – DANSE EU project – https://www.danse-ip.eu.
- Vehicles as constituents of a system of systems – Here, focus is on the constituent vehicles and on their architecture; in order to enable a vehicle to be part of the system of systems. Vehicle (architectures) should be engineered to enable connection and cooperation with other vehicles, roads, clouds, pedestrians, etc. There is not much work on this aspect and this is the main contribution of this paper.

In traditional vehicles, the car uses data locally stored in the vehicle and the communication based on signals between the different ECU (Electronic Control Units) of the car. In autonomous vehicles, the control can rely on aggregated data coming from multiple sensors but also from the infrastructure outside the vehicle such as the cloud. The communication rather than being signal-based within the vehicle it is service/IP based with the outside world. Sharing and controlling sensitive information could be of crucial importance in dangerous situations. For example, recently Volvo Cars have developed a mechanism to inform road users and road maintenance of slippery roads [46]. When a car detects slippery conditions on a road, it sends the information to the Volvo Cloud. This information can then be used to predict the road condition for later times [44]. When another car is approaching the slippery part of the road Volvo Cloud notifies the approaching vehicle that will automatically reduce its speed.

[8] See, for example, the IEEE Conferences on Systems Engineering (http://www.ieeesyscon.org), INCOSE (http://www.incose.org), the IEEE Systems Engineering Journal and the Journal of System of Systems Engineering, and the IEEE Systems of Systems Engineering – SoSE conference (http://sosengineering.org/2019/).

[9] Examples of projects are: COMPASS – http://www.compass-research.eu, DANSE – https://www.danse-ip.eu, T-AREA-SoS – https://www.tareasos.eu, ROAD2SOS – http://www.road2sos-project.eu/cms/front_content.php, CPSOS – http://www.cpsos.eu, AMADEOS – http://amadeos-project.eu, LOCAL4GLOBAL – http://local4global-fp7.eu, and DYMASOS – http://www.dymasos.eu.

3.2. Need of autonomy

As mentioned at the end of the previous section, in order to be part of an SoS, vehicles need to have some level of autonomy, according to the levels described in Section 2. In fact, in an SoS, constituent systems need to give priority to the goal of the SoS when the conditions of a scenario of the SoS are met. For example, see the description of the directed category of SoSs provided at the beginning of this section: the normal operation of constituent systems is subordinated to the centrally managed objective. This implies that the vehicle, sometimes within tight time constraints, needs to react autonomously and perform the required actions. To better understand we refer to specific examples: In Section 5.1.3, we describe a vehicle platooning scenario, in which various vehicles cooperate with each other to travel to a common destination. This scenario requires a timely communication and synchronization among vehicles and required reaction, such as braking, steering, accelerating, etc., in order to guarantee a safe and effective platooning. Similar discussion can be made for the cooperative collision avoidance scenario described in Section 5.1.4.

4. Electrical and software architecture

The increasing complexity of modern cars is asking for architecture descriptions able to provide the instruments to manage the development, which involves various actors of the automotive ecosystem. Proper architecture descriptions permit engineers to compare and relate different products across different vehicle programs, development units, and organizations.

For large-scale systems, such as in the automotive domain, a consistent focus on architecture throughout development enables the risk management and responsiveness to the stakeholder's needs of an agile development approach [43]. However, as testified by some studies [24,19,45], in practice there is often a discrepancy between the *as-intended architecture* – the architecture that captures the design decisions made prior to the system's construction – and the *as-implemented architecture* – the architecture that describes how the system has been built. In literature, this discrepancy between as-intended and the as-implemented architecture is called architecture degradation. The consequence is that the architecture description partially loses its benefits and reuse among different programs in the product line can be also compromised.

The work in [68] analyses the problem of assuring the conformance between multiple architecture descriptions and between architecture descriptions and code. This paper reports two surveys with 93 and 72 participants to examine architectural inconsistencies. The work suggests guidelines to limit the upfront architecture to stable decisions while paying attention to concerns that matter across team borders. The work in [69] focuses on understanding what parts of the architecture can be managed in an agile and flexible way and for which parts more controlled mechanisms are beneficial. In particular, the paper focuses on architectural interfaces in the automotive domain, since interfaces are key entities in determining and regulating the exchange of information between components, subsystems, and systems; in some sense they contribute in establishing boundaries (i.e. interfaces can be considered boundary objects [70,71]) between parts of the software system thus enabling agile teams to develop software or systems while maintaining a sufficient degree of autonomy.

The paper in [4] presents a functional reference architecture for autonomous driving, i.e. the architecture focuses exclusively on autonomous driving and specifically on the functional architecture, without considering the logical and technical architecture [12]. When there is the need for representing multiple aspects, then there is the need for a more structured approach. An architecture framework and its multiple viewpoints [28,45,12], is the right instrument to manage the increasing complexity of modern vehicles. An architecture framework is the instrument for (i) establishing a common practice for creating, analyzing, interpreting, and using architecture descriptions within the automotive domain and community of stakeholders, (ii) facilitating communication, commitments, and interoperation across multiple organizations, and (iii) enabling the development of architecture modeling tools and architecting methods.

According to the ISO/IEC/IEEE 42010:2011 standard [28] and as shown in Fig. 1, an architecture framework is composed of architecture viewpoints, which are used to organize an architecture description into architecture views. An architecture viewpoint is used to establish notations, conventions, techniques, and methods, which frame particular concerns and are conceived for specific system stakeholders. An architecture viewpoint is used according to specific model kinds.

The work in [45] describes an effort to establish an architecture framework for Volvo Cars to cope with the complexity and needs of present and future vehicles. The framework is based on the conceptual foundations provided by the ISO/IEC/IEEE 42010:2011 standard [28] and focuses on three new viewpoints that need to be taken into account for future architectural decisions: Continuous Integration and Deployment, Ecosystem and Transparency, and the car as a constituent of a System of Systems. Based on this experience, nowadays, Volvo cars is developing its own architecture framework.

In the context of the architecture framework, in this paper, we focus on the viewpoint that provides the description of cars as a constituent of a System of Systems.

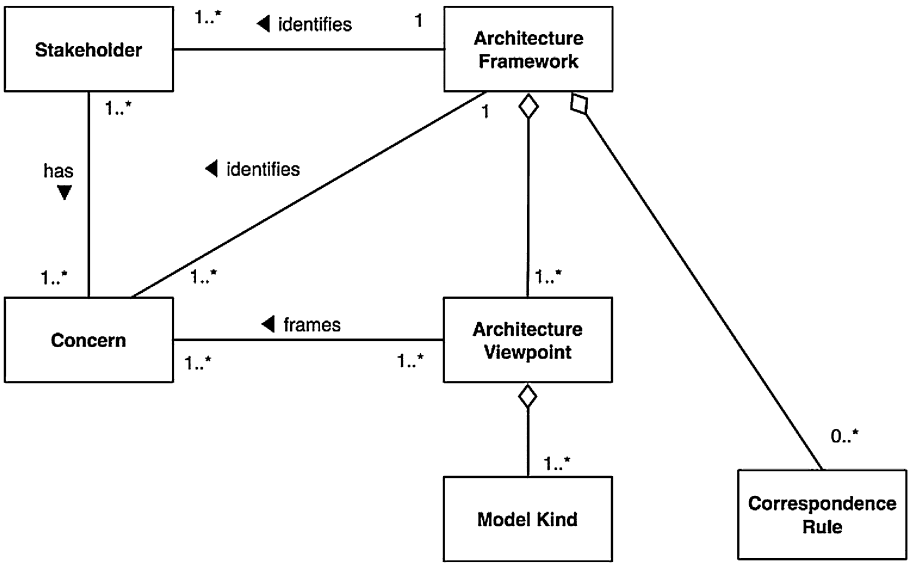

Fig. 1. ISO/IEC/IEEE 42010: contents of an architecture framework (figure taken from [28]).

5. System of Systems viewpoint

The structure of this section is based on the template called Architecture Viewpoint Template for specifying architecture viewpoints in accordance with ISO/IEC/IEEE 42010:2011, Systems and software engineering – Architecture description.[10]

5.1. Scenarios

In this section, we describe some motivating scenarios that have been used to identify the concerns. Additional SoS scenarios that we will probably see in the near future include, e.g., (i) improving traffic flow management by optimizing traffic lights and speed limits and by offering rerouting suggestions based on real-time traffic jam alerts, (ii) changing the color of street-lights to signal slippery road-sections or dangerous road conditions, when this is detected, e.g., by a connected car. In addition to that we mention a scenario foreseen by Klas Bendrik, Vice President and Group CIO at Volvo Cars Group (http://goo.gl/LuOHkz.): "*Imagine a world where road status data collected by cars is shared with other road users and with local authorities through a connected car cloud such as the Volvo Cloud: A world where the benefits of anonymized data-sharing support convenience and life-saving services while helping to contribute to a better society. Volvo Cars is working on realizing such a future scenario*".

5.1.1. Parking place search

There are several quite simple SoS services that can benefit from being implemented and coordinated by a central server in the cloud. The following parking place finder and reservation service is an example of such a service. A car that can assist to find free parking spaces, using equipped camera or radar, for example, can provide extra benefits to the service. Alternatively, most can be done just via smart-phone app integration.

A car driver wants to find a free parking space in the current location area. To do this the driver enables the service and information about the car's location is then sent to and used by a server to find free parking slots in the closest surroundings of a given location. When the driver has chosen a proposed parking space, it is reserved for a short time such that the driver can go there and register to use the parking slot (or else decide to let it free and continue searching). The server can then automatically charge the users according to the time for which the parking slots were used.

From the individual car as a constituent of an SoS viewpoint, the above example contains two to three main systems or stakeholders. These are: (i) the car-driver, (ii) the parking space resource manager and (iii) potentially a separate charging server. Besides these main systems, there is an obvious relation and dependency on other cars in the location area. In some sense, they are part of the same SoS but they can also be seen as being constituent systems in other SoS of the same type. In either of these cases there is a dependency among the cars that the coordinating parking slot resource allocator must be able to handle. Thus, it seems to be natural to view the SoS as a system that resolves parking problems in a location area for all the cars in the area that have the service enabled at a given moment in time.

[10] The template is called Architecture Viewpoint Template, which is released under copyright © 2012-2014 by Rich Hilliard. The template is licensed under a Creative Commons Attribution 3.0 Unported License. The terms of use are here: http://creativecommons.org/licenses/by/3.0/.

5.1.2. Ice on the road

On icy roads, vehicles are exposed to dynamic forces, which can lead to loss of road grip control and incidents. Report from Norwegian Center for Transport Research (TØI) [33] states that skidding or rollover by heavy goods vehicles in twisty road stretches often is the trigger of head-on collisions. Many single accidents with trucks also have the same trigger mechanisms.

Newer vehicles, both light and heavy ones, have electronic stability systems installed to avoid skidding or rollover situations. Anti-skid and anti-rollover-systems (ESC and RSS, respectively) are the main contributors to the positive trend in decreasing numbers of accidents. However, those systems can currently not prevent vehicles to enter curves in speeds exceeding what is physically possible for vehicles to manage. Since road conditions vary over time, critical speed also varies over time. Furthermore, every vehicle has its own constraints and does not correspond to fixed speed limits. The solutions to that problem from the SoS perspective are still under development, and one solution is proposed in [33]. Below we sketch one possible approach.

Vehicles, if equipped with sufficiently reliable slippery road analysis systems, can be used to detect and report to a central server about icy and other slippery road conditions along a road segment of a measured length. The server collates the incoming reports from more than one car. Then, when sufficient information has been gathered, the server can warn cars in the area about those road segments. How to handle the potentially varying degrees of confidence of the incoming reports, and when to remove or reduce severity levels of warnings, are open questions in need of further study. The server may also evaluate the confidence and trust of the individual reporting cars. Cars that contribute are supposed to get some benefits from the service provider, for example, get the service for free.

5.1.3. Vehicle platooning

A scenario of system of systems in cooperative future transportation systems is *vehicle platooning* (or road trains). Platooning can be seen as a collection of vehicles that cooperate with each other to reach some common goal, such as traveling to a certain common destination. The platoon is led and coordinated by a leading vehicle, manually or automatically driven. The following vehicles in the platoon follow the leading vehicle. Longitudinal and lateral control can be automated in the following vehicles. Some maneuvers, such as driving with short inter-vehicle gaps and joining the platoon from the side, may imply that a human driver is not capable enough and, therefore, control and coordination must hence be automated by the platoon. Cooperative Adaptive Cruise Control (CACC) is similar to platooning, but may have less coordination between vehicles and also less degree of automation, e.g. it may lack lateral automation. A brief overview of vehicle CACC and platooning systems is given in [8,62].

Key enabling technology for platooning is inter-vehicular communications. This type of communication is included in the concept of V2X-communication, i.e. (Vehicle to vehicle, infrastructure etc.). Platooning relies on cooperation among several vehicles and cannot be implemented with the use of local vehicular sensors only – its performance is only achievable through V2X communications. Platooning addresses the following societal needs:

- Cleaner transport. Truck platooning lowers fuel consumption and CO_2 emissions. When trucks can drive close together, the air-drag friction is reduced significantly. According to a recent study by ERTICO,[11] platooning can reduce CO_2 emissions by up to 16% from the trailing vehicles and by up to 8% from the lead vehicle. Scania reports an averaged figure of 12% [36].
- Safer transport. Truck platooning helps to improve safety since braking by the trucks following the lead one is automatic and V2X communications delays are shorter than human reaction times. Moreover, wireless communications allow enabling braking of all the platoon vehicles almost instantly instead of reacting one by one what prevents rear-end collisions.
- Efficient transport. Platooning optimizes transportation by using roads more effectively, delivering goods faster and reducing traffic jams. In certain situations, the driving range of trucks can also be extended.

Best current practice is that all platooning trucks are from the same brand. Platooning using European ITS-G5 standards to be able to platoon trucks of different brands is under development by ongoing VINNOVA[12] project Sweden4Platooning [3]. Further improvements are possible with the focus on the use of future 5G cellular infrastructure instead. Stockholm will be one of the first cities to have 5G technology rolled out already in 2018 by TeliaSonera and Ericsson.[13] The adoption of vehicle platooning contributes to the long-term competitiveness of Sweden, not only by supporting strategically important autonomous driving application, but also by developing forefront use-case to utilize this new 5G infrastructure.

According to the EU Roadmap for truck platooning technology by the European Automobile Manufacturers Association, standardization of communication protocols for platooning should be finished in 2021.[14] Regulatory changes and enabling policy measures required for platooning should be ready by 2023; which opens the possibility for market introduction. The

[11] http://erticonetwork.com/wp-content/uploads/2016/09/ITS4CV-Report-final-2016-09-09.pdf.
[12] https://www.vinnova.se/.
[13] https://www.ericsson.com/en/press-releases/2018/12/swedens-first-5g-network-is-live-at-kth-royal-institute-of-technology.
[14] https://www.acea.be/publications/article/infographic-eu-roadmap-for-truck-platooning.

state-of-the-art platooning technology belongs to Society of Automotive Engineers (SAE) automation level 2: the driving automation system performs both longitudinal and lateral vehicle motion control simultaneously, but all the drivers must monitor the driving environment, see [50]. Further evolution of platooning by 2030 will be to introduce SAE automation levels up to 4; allowing the driver of a trailing truck e.g. to rest. Also, platoons will operate in more complex environments such as sub-urban roads and smart cities, i.e. expanding the operational design domain (ODD). This will require further evolution of V2X communication protocols and is a subject matter of ongoing research in this direction.

Platooning stakeholders in Sweden include truck manufacturers (Volvo Trucks and Scania), telecommunications manufacturer (Ericsson), European Telecommunications Standards Institute (ETSI), 5G Automotive Association (5GAA), road traffic authorities, road operators, ITS Sweden, etc.

There is tight coupling between the performance of V2X communications and resulting safety of platooning. Especially challenging situation is an emergency braking, where the coordination is of utmost importance. The overall motivation is to avoid collisions within the platoon while still performing needed speed changes (acceleration and braking) as efficiently as possible. More research work is needed to characterize platooning as the SoS and to understand how V2X communication packet losses and communication delays impact safe headway times.

A dedicated emergency braking protocol aims to improve the outcome in an emergency brake scenario and safely enable smaller inter-vehicle gaps. Using only local sensors with no inter-vehicular communication to detect/signal e-brake leads to lower brake capability. A platoon braking scenario is described in [7]. Here two options are tested: braking based on standard V2X messages (CAM/DENM) [21,20], and braking based on an improved protocol. It is shown that legacy approach offers lower performance since it lacks coordination among vehicles and lacks data that are important for braking.

5.1.4. Cooperative collision avoidance

Cooperation and sharing of information in traffic can help to avoid accidents by supporting driver with information, warnings and even intervention when necessary. Preferred use-cases are to avoid getting in critical situations by adapting speeds, safety distances, or route selection strategies. Causes for many critical situations are described by the ITS-G5 standard from ETSI [20]: for example, slow vehicles, upcoming traffic jams, road repair works, hefty rain, slippery road conditions, or accidents. Other use-cases in critical situations, which require low latency, are described by the IEEE WAVE standard [60] such as Intersection Movement Assist, Left Turn Assist, Forward Collision Warning, Electronic Emergency Brake Lights [63], Lane Change Warning, and Control Loss warning.

There could be also new use-cases developed and used for further reducing collision risks; this can be achieved through new communication messages and protocols. Potential solutions to limit or avoid a potential upcoming collision from behind might lead to increase, rather than decrease, the vehicle speed.

Protocols that support coordinated lane change maneuvers are also foreseeable. It can also be imagined that sensors from vehicles could communicate trajectories of vulnerable road users to other vehicles that can cause collisions. This is called collaborative safety. More cooperative actions require standardization and market penetration; this implies that it will take time to concretely see the benefits of these innovative functionalities.

5.2. Concerns and stakeholders

In [45] we identified, through dedicated meetings with software architects of Volvo Cars, the main stakeholders of the architecture framework. They fall into the following main groups of stakeholders, namely *End-user* (e.g. used of the car), *Customer* (Purchaser of a car or related service), *Management* (responsible for scheduling, long term quality, groups, departments, and budget), *Developer* (responsible for creating the electrical system, its architecture, and the necessary tools that test and integrate the various components), *Maintainer* (responsible for interacting with the electrical system throughout its lifetime) of the electrical system, and *Specialist* (support for developers and maintainers on specific topics). Table 1 describes the various stakeholders and maps them to the group of stakeholders.

A further analysis is made together with architects working at Volvo cars, but also with architects and engineers working at suppliers to Volvo cars. Here we enriched the set of stakeholders by including (i) standard authorities for what concerns both safety and communication means, (ii) road authorities, (iii) Vulnerable Road Users (VRU), e.g. pedestrians, cyclists, and powered two-wheelers, (iv) IoT[15] devices (Internet of Things), which include devices to connect cars to smart buildings, smart lampposts, smart parking slots, etc. and (v) other OEMs, with the aim of engineering SoS in collaboration.

When engineering cars as constituents of an SoS, and then when a vehicle is connected and is part of an SoS, we can identify the following concerns[16]:

- *How to guarantee functional safety requirements when the car is part of an SoS, and;*
- *Once the car is part of an SoS, what are the implication on system design and functional distribution for functional safety?* Since the vehicle is connected and it is part of the SoS, it can use devices and information coming from different vehicles and/or other constituent systems. This will make more difficult to guarantee that safety requirements are satisfied.

[15] https://en.wikipedia.org/wiki/Internet_of_things.

[16] It can be noted that many or even all of these concerns are cross-cutting issues, i.e. have implications in more than one dimension. An example is the trade-of between identifyability of data and privacy, or safety vs. security.

Table 1
Overview of stakeholders.

Stakeholder	Group	Comment
Passengers	End-user	Passengers of the car
Drivers	End-user	Drivers of the car
Customers	Customer	Purchaser of a car or related service
Product planner	Customer	Acquirer of electrical system
Purchaser	Customer	Purchasers of electrical system
Line managers	Management	Responsible for scheduling, long term quality responsibility
Project managers	Management	Owns budget for development
System architects	Developer	Architects of the car
Functional developers	Developer	Owns functional and non-functional aspects (Synonyms: function owner; function realizer; function developer, function realizer, system developer)
Component developers	Developer	Developers of the components
SW supplier (internal/external)	Developer	Can be internal or external from the perspective of the OEM.
HW supplier (internal/external)	Developer	Can be internal or external from the perspective of the OEM.
Testers	Developer	Testers of the car
Attribute owners	Developer	Owns non-functional attributes such as performance
Tool engineers	Developer	Specifically testing tools, including design tools (e.g. for requirements)
Calibrators	Developer	Calibrators of the car
Diagnostic method engineers	Maintainer	Engineers of the diagnostic method
Workshop personnel	Maintainer	Personnel working in the workshop
Fault tracing specialists	Maintainer	Specialists of the fault tracing
Technical specialist	Specialist	Support developers and maintainers on specific topics
Safety authorities	Standard	To guarantee compliance to safety regulations
Communication authorities	Standard	To guarantee compliance to communication regulations
Road authorities	Road	Connection with the road infrastructure
Pedestrians, cyclists, powered two wheelers	VRU	Connection with pedestrians, cyclists, and powered two wheelers, and safety guarantees
IoT specialists	IoT devices	Connection with smart buildings, smart lampposts, smart parking slots, etc.
Other OEM representatives	Other OEMs	Definition and engineering of SoSs and agreement on ownership, management, evolution, etc.

- *Once functional safety requirements involve devices that are outside of the vehicle (other constituent systems of the SoS), how to ensure that these requirements will be guaranteed?*
- *How the methods and processes for end-to-end function development and continuous delivery of software need to evolve?*
  Once the vehicle is part of an SoS, methods and processes for both end-to-end function development and continuous delivery of software need to be evolved.
- *How to establish and guarantee a reliable and efficient communication between the vehicle and heterogeneous entities, such as other vehicles, road signals, pedestrians, etc.?*
  One of the enablers for having a vehicle as part of an SoS is to have the technology to establish and guarantee a reliable and efficient communication between the vehicle and all the other devices and constituent systems, e.g. other vehicles, road signals, lampposts, pedestrians, cyclists, etc.
- *How to ensure that the vehicle and other constituent systems of the SoS will be able to exchange information and to correctly interpret and use the exchanged information?*
  Besides of guaranteeing an exchange of information from and to the vehicle, there is the need to ensure sufficient interoperability [55], i.e. that the constituent systems will be able to correctly interpret and make use of the data that have been exchanged.
- *Once the vehicle is connected, how to guarantee that the security of the vehicle is preserved?*
  In order to be part of an SoS, the vehicle needs to be connected and therefore, it is exposed to attacks that can compromise the expected behavior of the vehicle.
- *How to identify a balanced tradeoff between shared data and users' privacy?*
  The exchange of data is necessary in order to enable the definition of useful SoSs. However, collecting and sharing data can violate the privacy of users.
- *How to keep the data shared within the SoS – and potentially their replication – sufficiently updated or synchronized?*
  Data that is exchanged within the SoS need to be updated and reliable. Potential replicas of data need to be synchronized and in general data need to be consistent.
- *How to manage the age of available information?*
  Data shared within the SoS need to be fresh and the age of data need to be properly managed in order to guarantee that the available data will be always reliable and functional for the envisioned scenarios.
- *Which functions in the car are allowed to make use of data coming from other constituents?*
  Most probably only a specific subset of all the components and functionalities of the vehicle will be allowed to make use of data coming from other constituents. In fact, the challenge here is related to the quality and security of the data; The devices that are used to collect the data are outside of the vehicle and potentially unknown and uncontrollable.

The integrity of data needs to be specified for it to be of use in a safety-related function, the data trustworthiness, authorship and non repudiation of the data must be established, etc.

5.3. Definition of the model kinds

ISO/IEC/IEEE 42010:2011 [28] defines requirements on the description of system, software and enterprise architectures. It aims to standardize the practice of architecture description by defining standard terms, presenting a conceptual foundation for expressing, communicating and reviewing architectures and specifying requirements that apply to architecture descriptions, architecture frameworks and architecture description languages. According to the ISO/IEC/IEEE 42010 standard, in this subsection we identify the model kinds, i.e. conventions including languages, notations, modeling techniques, analytical methods and other operations. First, in Section 5.3.1, we identify the main concepts that should be considered when architecting cars as constituents of a future transportation system. Second, in Section 5.3.2, we describe the notations and tools that can be used to define one or more architecture models composing a specific architecture view. Section 7 will then show an architecture model, representing in particular the functional reference architecture for a car that would be part of an SoS.

5.3.1. Concepts that need to be covered

Distributed end-to-end functionality In the SoS scenarios described above, functionalities are not only referring to nodes that are within the car but also outside the car, such as cloud services, other vehicles, infrastructures, pedestrians, etc. A car in a System-of-Systems may receive information from sources outside the car itself, e.g., from other vehicles and road signals and road side based sensors. Sensor data may be used even though the cars are not operating cooperatively, i.e., the information are provided by remote sensors. Therefore, distributed end-to-end functionalities involve uncontrollable and unforeseeable actors, such as pedestrians, road signals, etc. Functionalities cannot be completely planned at design time since we do not know which nodes and actors will take part of the functionality. In SoS scenarios, more often than not, part of functionalities can only be realized at runtime, since they emerge from the collaboration of the constituent system and are observed at the boundaries of the SoS. It is then important for each single car to have enough information to validate the received data and be able to perform as good sensor fusion as possible as a base for tactical decisions. To this end, sensor data should be accompanied with meta data with information such as confidence levels. While collective perception described above enhances perception capabilities of vehicles, it also increases the load on the wireless channels. It is an open research problem to resolve a trade-off between the number of perceived objects which a vehicle communicates to others and the generation frequency of collective perception messages. Indeed, more objects are exchanged better the environmental awareness is, while latency between two consecutive messages impacts the object age and information accuracy [23]. In vehicular networks with a large number of nodes, the communications about all perceived objects with a high message generation frequency would result in channel congestion, packet losses, and increased latency.

Functional safety Connecting vehicles into SoS has far-reaching consequences on how to perform risk and safety analysis. In fact, every characteristic that distinguishes an SoS [38] has its implications on safety, as will be discussed in the following.

The **managerial and operational independence** of the constituent systems (CS) means that SoS-level safety requirements need to be analyzed in cooperation, and then agreed upon, by the different stakeholders that will be involved in an SoS. Also, the overall safety requirements need to be balanced with respect to the individual safety of the CSs.

Evolutionary development of the CSs, and as a consequence, of the SoS, means that both the safety analysis and the resulting agreements need to be constantly supervised and updated. How such supervision should be done is far from evident and will probably differ among SoS. It could for example be done by a dedicated high-level agent, it could be mediated by a third-party, or it could be done cooperatively by the CSs. The composition of SoS may also evolve over time, further complicating safety analysis.

Emergent behavior, is the main reason for constructing an SoS, but also becomes one of the main challenges when it comes to analyzing safety. Analyzing safety in scenarios in which there are interactions with external and independent systems is challenging. It can be argued that this is not sufficiently addressed by the ISO 26262 standard [26]. An obstacle is the definition of item of being a function at vehicle level; which can be interpreted to not be capable of spanning SoS. Previous work has investigated this e.g. [42]. Careful definition of the SoS-function may overcome this obstacle without the need for redefining the standard. Due to emergent behavior, a large portion of potential hazards can be attributed to incorrect or unforeseen interactions between systems or system components [47,35], and also there can be limited knowledge and control of external systems.

Geographical distribution means that CSs need to rely on communication links for the correct operation of an SoS. While the correctness of data from the other SoS participants is crucial, CSs should have defense mechanisms for the cases when data is incorrect. Thus, on the vehicle level it is important to handle that data may be e.g.: (i) unreliable – CSs can appear and disappear; (ii) untrustable – data may be corrupted or even subject to a malicious attack; (iii) ineffective – guarantees about performance, reliability, etc., may be missing.

In addition, due to the evolving nature of SoSs, together with a potential lack of transparency between OEMs, the SoS design can be only partially known at the time of safety analysis. Also, SoSs are typically socio-technical systems with machines and humans interacting in complex ways. Thus, it is important to take human factors into consideration, not

least including the human perception and expectation of how the SoS should operate, and also possibly unsafe human reactions to environmental events. Another trade-off that is important to consider is the balance between increased risk and functional requirements. For example, should a vehicle reduce its speed and lead to an inconvenience for its occupants, only to reduce the risk to other road users?

There is a need for runtime strategies for handling safety levels of services, and the difference between design time assessments and run time assessments. For example, how is conflicting data handled; that is usually, at least partly, redundant? An important aspect for the SoS is that sensor data transferred between different nodes in the SoS need to be accompanied with meta data that describe the quality of the data. For example, what is the confidence that there is no other car at a certain distance in front of the car with the sensor? Evolutionary machine learning may play a part as a robust method for the "sensor fusion" needed to find patterns emerging from a group of connected cars.

Connectivity and interoperability Sufficiently dependable connectivity is essential to enable the expected service level in different places in a future transportation system. Making connectivity sufficiently dependable may be possible through the use of several different channels such as through vehicular and cellular networks.

Currently, there are two complementary V2X communication technologies being developed: "ad hoc" ITS-G5/802.11 (no central coordination of communication) and "cellular" LTE/5G (infrastructure based coordination of communication). Currently, there is no dominant solution [59]. For the ad hoc case, ETSI has delivered the first ITS-G5 release of a set of vehicular communication standards in 5.9 GHz band under European Commission Mandate M/453. ITS-G5 specifies the overall communication stack including the IEEE 802.11p standard at the two lowest layers. The corresponding protocol stack in North America is specified by IEEE 1609.x WAVE. For the cellular case, LTE release 12 (2015) specifies device-to-device communications among users within the cellular network by introducing sidelink communications. LTE release 14 (2017) incorporates specific vehicular aspects (Cellular-V2X); specifically two modes that allow users to operate in coverage as well as out of coverage of the base station.

Moreover, there still has to be on-board functions that handle graceful degradation of services when connectivity is limited, delayed, or not available at all. Thus, regardless of the availability of the connectivity, the user shall experience a robust behavior of the functions; this is especially important for safety-related functions.

Interoperability is the *"ability of two or more (software) systems or components to (i) exchange information and (ii) use the information that has been exchanged"*, according to the definition provided by the ISO/IEC/IEEE 24765:2010 standard on Systems and software engineering – Vocabulary [27]. This general definition has been conjugated in many different ways based on the reference application area and on the many different factors and aspects characterizing them. Interoperability involves standards, protocols, and integration and adaptation of interfaces to enable the effective and efficient communication between CS. Interoperability is multifaceted, as clearly emerge from the INTERO interoperability evaluation framework, which enables the evaluation of products/systems/product lines from the point of view of interoperability and enables the identification of the specific interoperability dimensions on which organizations should act in order to improve interoperability [55].

When focusing on future transportation systems, interoperability can be defined as the ability of two or more CS that are part of the SoS to exchange information and to use the information that has been exchanged according to the goal of the SoS. The Levels of Conceptual Interoperability Model (LCIM) interoperability standard [66,58] proposes 7 interoperability levels. More details can be found in [55].

In the context of future transportation systems, unambiguous interpretation of shared data between systems is necessary for interoperation, but it is not sufficient. Despite standards for shared data that provide specification with the objective to enhance the functionality and interoperability, the data encoded using these standards are not necessarily interoperable. For instance, concepts that have the same labels, and somehow even the same meaning, can be used completely differently in different applications. This is for instance the case of the label "speed" within a car that can have different meanings in different applications or contexts unless the semantics is very clearly defined and acted on.

Security and privacy Connectivity of cars poses new challenges both on security and privacy since cars become exposed, as any other device that is connected to the Internet. Malicious software can potentially take complete control of the car and this opens to many problematic scenarios with potentially dramatic effects. Aspects that should be considered span from attacks to take the control of the car to attacks for spying and extracting information, trying to use the car for doing another attack, spoofing attacks, etc. Firewalls and DMZs are possible solutions, but there can also be some intrusion detection systems and more advanced solutions that can be distributed in the SoS and exploit the knowledge and expectations of constituent systems to identify and isolate constituent systems that are under attack.

There is an interesting trade-off between safety and security that should be considered. Sometimes these issues can correlate and require the same solution, e.g. fallback mechanisms in case of incorrect data, while in other cases they can be contradictory, e.g. when balancing communication latency and message encryption. Thus, it is important to consider safety and security in the same analysis, and also to do so early in the SoS development.

Especially critical safety problems pop up in cases when the vehicles use connectivity as a part of their control loops, e.g. in platooning scenario described in Section 5.1. Examples of security attacks that can compromise safety can be caused through:

- Falsifying multiple vehicle identities so that events can be generated by these false nodes to interfere with legitimate vehicles. Multiple identities may be used by an attacker to join a platoon, overloading the leader, which has to manage falsified members. Another scenario is the use of falsified members at strategic platoon locations, which collude to send erroneous beacons, potentially causing a road accident [11].
- Malicious jamming originating from a radio transmitter located in the vicinity of communicating vehicles. When located along the road, a reactive jammer can substantially increase the packet losses at inter-vehicular communications links of platooning vehicles, including complete blackouts for a couple of seconds. Since vehicles in platoons are moving with a few meters inter-vehicle gap, the detection of this situation should happen in real-time within a fraction of a second [37].

For what concerns privacy, cars will need to share information to enable the scenarios of SoS described in Section 5.1, however, sensitive information should be properly protected. The point is to find the right trade-off between data that is shared since it enables many interesting functionalities and data that should be kept private.

An important perspective is also understanding how to establish the chain of trust in a future transport system. How to authenticate that the received data has not been changed? An assumption is to use some type of third party certificates.

An example is information based on remote sensor data. How to ensure that the received data can be trusted, validate the received data, make sensor fusion with other external and internal sensor sources and base tactical decision on available data?

5.3.2. Tools and notations

This section described the tools and notations that can be profitably used to define architecture models for this viewpoint.

When focusing on safety and security, an interesting tool and notation can be identified in the SAFE metamodel produced in the ITEA EU project.[17] The metamodel is essentially an extension of EAST-ADL and AUTOSAR to enable integrated safety modeling and safety analysis. The metamodel integrates the artifacts associated with the application of the ISO 26262 standard and, thus, it covers the safety related elements and relationships necessary to ensure the safety requirements. More specifically, the metamodel integrates EAST-ADL (Electronics Architecture and Software Technology – Architecture Description Language),[18] Autosar, and ReqIF[19] (Requirements Interchange Format) used for the specification of the requirements. SAFE RTP is a reference technology platform for the SAFE project and it is an EMF-based Java implementation of the SAFE meta-model. SAFE RTP [65] integrates the AUTOSAR meta-model from ARTOP[20] and the EAST-ADL meta-model from EATOP.[21]

Safety and security on the architectural perspective, has been also investigated in the H2020 EU project SAFURE – Safety And Security By Design For Interconnected Mixed-Critical Cyber-Physical Systems.[22] The project aimed at providing a holistic approach to safety and security by construction. The SAFURE metamodel[23] contains modeling concepts for the representation of safety, security and time attributes and constraints in mixed-critical systems. These modeling concepts have been proposed in an abstract way so to enable a possible concrete implementation according to UML/SysML and AUTOSAR.

While SAFURE focuses on the security of in-vehicle communication and its relationship with performance and safety, the EVITA project[24] has been essentially focused on intra-platform security issues and privacy. EVITA proposes a metamodel to security and trust. Trust can have various levels and there are two main actors (e.g. a person, group, organization, or system who affects or can be affected by some actions): a trustor and a trustee. A trustee is an entity that provides trust for the use of a function or service whereas trustor is an entity that uses a function and has a justifiable belief that the trustee will provide the expected function or service.

For what concerns interoperability, we refer to the European Interoperability Reference Architecture (EIRA),[25] which has been defined to guide public administrations in their work to provide interoperable European public services to other public administrations, businesses and citizens. The suggested architectural style is the service-oriented architecture (SOA).[26]

However, it is important to highlight that the notations and tools to be used to produce an architecture description should be properly selected according to the purpose and the intended audience of a particular architectural model [24]. In some cases, an informal box and line notation would perfectly satisfy the goal of the architecture models, as also testified

[17] https://itea3.org/project/safe.html.
[18] http://www.maenad.eu/public/conceptpresentations/EAST-ADL_WhitePaper_M2.1.12.pdf.
[19] https://www.omg.org/spec/ReqIF/1.2/.
[20] Eclipse-based implementation of the AUTOSAR meta-model: www.artop.org.
[21] Eclipse platform implementation of EAST-ADL: www.eclipse.org/proposals/modeling.eatop.
[22] https://safure.eu.
[23] https://safure.eu/downloads/SAFURE-D2.1-PU-M12.pdf.
[24] FP7 EU project EVITA: https://www.evita-project.org/.
[25] https://joinup.ec.europa.eu/sites/default/files/distribution/access_url/2018-02/b1859b84-3e86-4e00-a5c4-d87913cdcc6f/EIRA_v2_1_0_Overview.pdf.
[26] https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm.

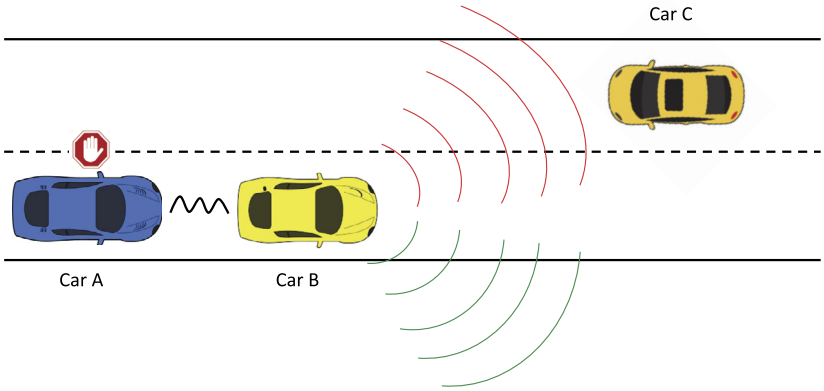

Fig. 2. Demonstrator overview.

in the study reported in [39]. This is actually the case of the reference architecture described in Section 7, where we show it by making use of a box and line notation.

5.4. Correspondence rules

This viewpoint has correspondences with various other viewpoints. First of all, this viewpoint has correspondence rules with the other SoS viewpoint that focuses on the overall SoS. As anticipated above, the other SoS viewpoint aims at giving an answer to the following question: "*How to engineer the SoS so that the collaboration among various constituent systems will achieve the SoS goals?*".

Then, this viewpoint has correspondences with the functional safety viewpoint since, on one side, having a vehicle connected might pose new challenges to functional safety. On the other side, it opens new opportunities related to safety that open new promising scenarios that are unthinkable for a disconnected vehicle, such as, the ice on the road, the vehicles platooning, and the cooperative collision avoidance scenarios; described in Section 5.1.

Moreover, having the vehicle connected opens new challenges in terms of privacy and security, as described above. This triggers correspondences with the security and privacy viewpoint.

Finally, this viewpoint has also correspondences with the topology, cost, variability, continuous integration and deployment and ecosystem and transparency.

6. Demonstrator

In this section we demonstrate the value of the SoS viewpoint described in this paper and consequently also of the architecture framework that has been proposed in [45]. Specifically, we base the validation on our experience of applying the SoS viewpoint to an *overtake* scenario. This is an additional scenario with respect to those described in Section 5.1 and it is described here in the following. In this scenario, three cars participate. As shown in Fig. 2, Car A is the overtaking car, Car B is the car that is overtaken, and Car C is an oncoming car that makes the overtaking maneuver risky. Car A and Car B engage in car-to-car communication, whilst Car C is not equipped with such equipment leaving Car C silent. As soon as the driver of Car A indicates her goal to overtake, Car B will share information of its sensors. This will help to issue a do-not-pass-warning, even though the sensors of Car A have no way of seeing the oncoming Car C. In this way, Car B is extending the sensor range of Car A.

This scenario has been demonstrated as part of the NGEA project, showing different triggers for indicating an imminent overtaking action. The demonstrator was based on a single Tier-1 supplier's technology stack integrated into three cars of the same brand. While this demonstrator is a milestone towards making such technology available, it is only a first step towards engineering a system-of-systems that can effectively make overtaking safer.

During a workshop with representatives of several participating companies, we applied the stakeholder concerns from our SoS viewpoint (see Table 2). We cannot share technical details regarding the answer to the questions, since such details relate to current product development of partner companies. It is also out of scope, since we aim to validate the SoS architectural viewpoint, not the take-over scenario. Still, the workshop allows us to present an argument for each to show its relevance and usefulness in Table 2).

7. Functional reference architecture for a car as a constituent of an SoS

This section describes a reference architecture for the vehicles that are constituent systems of an SoS. This reference architecture is a generalized architecture of a solution, which based on best-practices, experience, and knowledge acquired in our consolidated collaboration with various companies, as well as study of relevant literature.

Table 2
Validation of concerns from SoS viewpoint.

Concern	Validation argument
How to guarantee functional safety requirements when the car is part of an SoS?	The overtake scenario clearly refers to safety-critical functions. Issuing a false warning, or not issuing a warning when over-taking is dangerous, could endanger the vehicles and their occupants. Functional safety depends on the interplay of Car A and Car B.
Once the car is part of an SoS, what are the implication on system design and functional distribution for functional safety?	It is clear that it is the responsibility of Car A to determine whether over-taking is safe. For that, there should be proper interoperability between Car A and Car B, and Car A must establish that the data received from Car B is sufficiently accurate and timely.
Once functional safety requirements involve devices that are outside of the vehicle (other constituent systems of the SoS), how to ensure that these requirements will be guaranteed?	This question points towards the need to enable reasoning about safety and data quality at runtime. Protocols and interfaces as well as reasoning about trust will become important areas. It is in fact impossible to assess safety concerns once and for all and to relate them to the sensors and devices of a car, in this case Car A. As we said before, Car A is extended by the devices in Car B. In other similar scenarios Car A could be extended with devices of other cars, depending on when another instance of the overtake scenario would happen. This implies also that the evaluation of "trustable" vehicle should be done by the car that want to overtake, in this case Car A, when another vehicle that is functional to the realization of the scenario is met.
How the methods and processes for end-to-end function development and continuous delivery of software need to evolve to be suitable in a System of Systems setting?	Pushing an update or a new feature to one of the vehicles might compromise the interoperability of the vehicles. In this sense, the delivery of new software to the car should seriously take into account not only the functionalities of the isolated vehicle but also the SoS scenarios that a vehicle is engineered for serving as a constituent of an SoS.
How to enable a reliable and efficient communication between the vehicle and heterogeneous entities, such as other vehicles, road signals, pedestrians, etc.?	While technical feasibility has been shown in the demonstrator, it remains unclear to what extend this can be achieved with various technology stacks by different vendors. Aspect of disturbances, adverse weather conditions, etc., should be also taken into account.
How to ensure that the vehicle and other constituent systems of the SoS will be able to exchange information, correctly interpret and to use the information that has been exchanged?	While the technical feasibility of the exchange has been shown, the usage depends on advanced standards and protocols that allow for runtime reasoning. Also, this is a good example to show that exchanging data is not enough; there is need for proper interoperability. Referring to Section 5.3.1, we need a communication protocol, and the connected vehicles need to be able to communicate. However, they should be also able to properly understand and interpret the exchanged messages (semantic interoperability). Moreover, the vehicles will behave assuming that the exchanged data will have the effects they expect on the other vehicle (pragmatic interoperability). Time is also an important aspect since Car A would need timely data from Car B and data soon become obsolete (dynamic interoperability). This scenario is not so complex from the point of view of interoperability, since it basically only requires some data sent from Car B to Car A when Car A asks for that. In this sense, a level of conceptual interoperability is probably not necessary.
How to guarantee that the security of the vehicle is preserved once the vehicle becomes connected?	This question was deemed relevant during the workshop, requiring advanced protocols that allow to verify and compute trust in data.
How to identify a balanced tradeoff between shared data and users' privacy?	This question points towards constraints for the solution space of the questions above, since it makes solutions favorable that do not store the id of the individual cars.
How to keep the data shared within the SoS (and possible replication of data) sufficiently updated or synchronized?	This concern was considered relevant, since it requires the consideration of recording critical information.
How to manage the age of available information?	The take-over scenario requires current information. As discussed above this is important for having a proper interoperability among the vehicles, specifically for having dynamic interoperability.
Which functions in the car are allowed to make use of data coming from other constituents?	This concern was regarded crucial for designing the SoS. Clearly, a specialized component will be required to collect external data and to judge whether it is trustworthy and has sufficient quality for different usage within the car.

The goal of this reference architecture is to assist architects of connected vehicles by raising various considerations and solution possibilities that influence the architecture of a vehicle that is connected and aims at being a constituent system of an SoS. This reference architecture can be considered a common ancestor for a range of various implementations [4], where each particular solution would be identified by particular decisions that will be made in the context of specific projects and by properly considering pros and cons, and tradeoff. This reference architecture is based on the architecture viewpoint proposed in Section 5.

Fig. 3 shows the functional reference architecture. According to [4], we use the term functional architecture to refer to the logical decomposition of the system into components and sub-components, and the data-flows between them. This reference architecture does not describe the dependencies on specific implementation technologies. The reference architecture is inspired by the reference functional architecture for autonomous driving, which is proposed in [4].

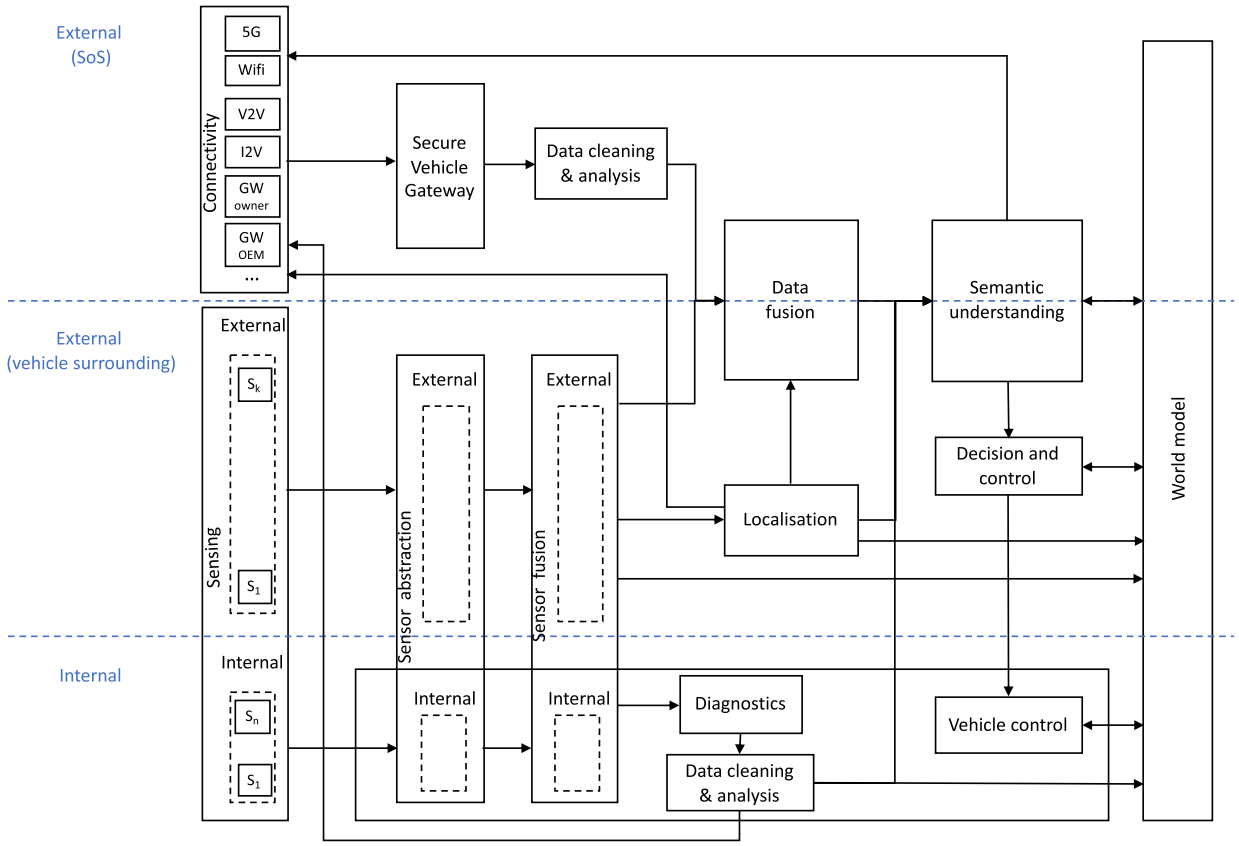

Fig. 3. Functional reference architecture for vehicles that are constituent systems of an SoS.

As shown in the figure, we identified three different levels that are the three different "perception" capabilities of a vehicle. Internal refers to the sensing capabilities within the vehicle. External (vehicle surrounding) refers to the capability of the vehicle to sense the surrounding with, e.g., instance camera, radars, and lidar. The third level refers to the external (SoS) that enables the vehicle to receive information beyond the limits of its sensors and devices, since it refers to information received from other vehicles, street signs, parking, pedestrians, etc., via various communication means.

Focusing on the internal level, through a sensor abstraction, data received from multiple sensors are taken as input from the sensor fusion component, which will make sense of them and produce the needed knowledge. Diagnostic and the data cleaning and analysis components will clean the data and forward them to the world model, which holds the state of the external environment and of the vehicle itself within this environment, and to the semantic understanding component, which will be explained later.

Sensor abstraction and sensor fusion work in similar way for sensors and devices that are sensing outside the vehicle. This information is used by the localization component, which is able to determine, with the needed accuracy, the position of the vehicle with respect to a global map. Indeed, the data produced at this level are also used to inform and update the world model.

The external (SoS) level is taking information via the connectivity component, which includes various communication means, such as 5G, Wifi connection, V2X (This collective term includes V2V – Vehicle to Vehicle, V2I – Vehicle to Infrastructure etc., and can potentially be the same technology) communication, Gateway (GW) of both the owner and the OEM, etc. As described in Section 5 this information is coming from devices belonging to third party, out of the control of the vehicle, potentially unreliable and untrustable. This is why these data require special attention. First of all these data are managed and filtered by a secure vehicle gateway component, which might act as a firewall, but that can also provide intrusion detection functionalities. Moreover, this component would also include advanced techniques to evaluate the trustability of the source of information. Data that are considered secured and trusted enough are further cleaned and analyzed. Then these data are "fused" with data coming from the external sensors and devices of the vehicle. The semantic understanding component has the difficult role of finding the "right" understanding of information coming from various sources of information both internal and external of the vehicle. The component needs to solve also conflicting information and send to the decision and control component the required information in order to take a decision. The decision and control component is the component that will instruct the vehicle control component to take all the actions needed, for instance, to play the expected role of the vehicle in the SoS. This component might decide that the vehicle should temporarily give priority to the

actions that are needed by the SoS. As discussed in Section 3.2, this component highlights that in order to have a vehicle as part of an SoS there is the need of autonomy. Therefore, this component will need to coordinate with the components managing the behavior of the vehicle when acting as an independent entity. This highlights also the strong connection that should be established between this reference architecture and the reference architecture for autonomous vehicles, such as the one proposed in [4].

We defined and evaluated the reference architecture internally to the project. We also contacted a safety engineer external to the paper to further evaluate the reference architecture. The safety engineer has 25 years of experience in system safety in the aerospace and automotive domain, and is currently working as system and system safety architect for an OEM. The assignment is to design an electrical platform and architecture for ADS that actually works – instead of evolving the current deficient vehicle platforms and failing, i.e. not achieving safety. We presented the reference architecture as shown in Fig. 3 and provided some explanation. The following comments were given:

- *This logical view is a "first" attempt that does not take any impression of the effects of allocation, such as the identification of required safety-mechanisms as found in safety analysis. An effect could be the realization that additional redundancy is needed. Also a required level of integrity may not be reachable without redundancy even at this logical level. Currently there is no redundancy in Fig. 3.* This, and other factors such as not knowing the function that will use the platform, implies that we cannot really make any judgment on safety given the current information. This comment highlights that this particular architecture model needs to be complemented with other architecture models in this viewpoint as well as in other viewpoints. However, for separation of concerns we believe that it is better to do not include these aspects in the reference architecture and, instead, to exploit mechanisms, like the correspondence rules in Section 5.4 to create the needed connections among the various architecture models and views.
- *Generally there is much external information from other systems (map-data, objects that have been identified in other systems etc.) that could be of use in a local system. However, how can it be established that this information is trustworthy and is available at all?* We completely agree with this comment and this aspect is crucial. This is exactly the role of the Secure Vehicle Gateway and Data Cleaning & analysis components, as described above in this section. We provided a proper discussion of these aspects to avoid misunderstanding in the readers as it happened with our safety expert.
- *In the ITS-G5 standard (and most probably the other connectivity services) there is support for security, authenticity, encryption, privacy, etc. However, how can it be ensured that the original source of information, e.g. a sensor, is correct?* As we discussed above, proper techniques should be used to guarantee that evaluate the trustability of the source information. These techniques will be deployed into the Secure Vehicle Gateway component. We can imagine and recommend to use techniques like blockchains [76] or Information-Flow Control [49]. Blockchains can be employed to secure personal data against tempering and revision [76]. A blockchain consists of data-structure blocks stored in a decentralized architecture consisting of potentially infinite number of nodes. Each node has a copy of the blockchain. Whenever new transactions occur, the consensus of all the nodes is needed in order to add the new block into the blockchain. Thus, each transaction is held by the entire network and if someone attempts to cheat the system, she can be easily identified. Information-Flow Control (IFC) [49] is a promising technology for the active data management. IFC permits to obtain guarantees of many of the GDPR requirements related to how private information gets handled and disseminated within systems. IFC was originally conceived to confine sensitive information in operating systems [5,74,32] but it has also been used to secure web browsers [1,56] and many programming languages [40,48].
- *In ISO 26262 there [is] a possibility to perform decomposition of a high integrity/ASIL function to multiple components with lower ASIL. E.g. ASIL D is decomposed to a small part with ASIL D (D) and another part with QM(D). Observe the parenthesis notations which indicates the original (highest) ASIL. Note also that integration of these two decomposed parts is still done at the higher ASIL, i.e. ASIL D. Decomposition can be done several times and at different architectural views. QM implies that the function is not safety critical and does not need to be developed with ISO 26262; and development according to a "normal" quality management system (e.g. IATF 16949:2016) suffices. However, it must be stressed that we cannot compose multiple external QM-sources (by combining e.g. through data fusion) to reach any notion of integrity, i.e. ASIL. The conclusion is therefore that a function can definitely be made to "work better" in terms of performance, but in terms of safety, the information cannot be used unless it is qualified with some level of integrity. This includes the entire chain from e.g. source, transmission of the data (i.e. network), destination etc. Today, it is still a challenge to show that the source is "correct".* This comment highlights what we discuss largely in this paper, especially in Section 5.3.1. This comment requires no change to the architecture but we report this comment since it is important to highlight this important aspect from the safety point of view.

8. Engineering an SoS: challenges and opportunities

Connected vehicles will combine data from inside vehicle with external data coming from the environment (other vehicles, the road, signs, and the cloud). Future transportation systems will be a heterogeneous mix of items with varying connectivity and interoperability. A mix of new technologies and legacy systems will co-exist to realize a variety of scenarios involving not only connected vehicles but also smart roads, pedestrians, cyclists, and so on. In other words, future transportation systems are expected to be system of systems, where each constituent system can act as a standalone system, but where the cooperation among the constituent systems will enable new emerging and promising scenarios. In today's vehicles, the driver plays a fundamental role. Besides controlling the vehicle, she/he is also expected to recover the vehicle

from failures. With fully autonomy (SAE level 5), the driver is no longer in the loop. The vehicle, besides controlling the basic driving maneuvers, should handle all possible situations during driving. As the driver is not monitoring the vehicle and is not a fallback option anymore as in the lower levels of automation, it is expected that nothing goes wrong. For the higher levels of automation (i.e., levels 3-5), additional players enter the ecosystem: companies that focus on communication aspects will play a big role in this ecosystem, as communication will be an important basis for highly automated driving [31]. Communication allows the vehicles to ensure that they have redundant information sources (i.e., several sources of information) to ensure safety, reliability, and quality of autonomous vehicles. In other words, autonomous cars are becoming part of an autonomous system of systems. An autonomous car needs to constantly communicate with other cars and the infrastructure, via sensors and the cloud. As foreseen by IBM, unprecedented complexity will lead to unprecedented cooperation.[27] This is also confirmed by some discussions we had with several Swedish companies (between others Volvo Car Group, Volvo Trucks, Ericsson, Axis communications, Jeppesen – Boing); all of them claimed that one of the business challenge for industry in the near future is to build and provide new functionalities and/or services in a short time scale by easily integrating the system with other systems and to continuously maintain and evolve the system under guarantees of quality of service.

As anticipated in Section 3, SoSE is an emerging discipline of the last years that addresses the development, operation and maintenance of SoSs. Even though the community around SoSE is active, SoSE is a young discipline, and general principles and theoretical foundations remain to be discovered. Engineering SoSs requires cooperation among companies, and there is the need of standards. This is testified by the investigation of Toyota about intelligent transport systems[28] and by the pn-European project SOCRATES2.0 (System of Coordinated Roadside and Automotive Services for Traffic Efficiency and Safety),[29] which aims at setting new standards to share and integrate traffic information to enable effective traffic management and navigation services.

Once defined a collaboration among the players of the system of systems, many questions need to be answered, e.g.:

- Should the SoS be Virtual, Collaborative, Acknowledged, or Directed, or a mix of them?
- What is the extent of “controllability” of the constituent systems?
- Should the collaboration be established for a limited time in response to or to solve specific situations, such as hazards in connected safety scenarios?
- Should the collaboration be extemporary and spontaneous?
- Who is engineering the SoS, one specific OEM, a group of OEMs, road authorities, etc.?
- Who is the owner of the SoS?
- Who is the manager of the SoS?

Moreover, some of the scenarios that are supported by the SoS can be safety-critical. Consequently, SoS engineering needs to be grounded on a rigorous conceptual framework, so to enable: (i) precise and unambiguous definition of SoS properties; (ii) runtime verification so to control, avoid, or block negative/harmful unanticipated emergent behaviors (emergent behaviors will be explained later in this section), and (iii) the development of tools for managing the SoS. In order to understand the complexity of the problem, we mention two challenging characteristics of SoSs:

- Uncertainty about constituents – SoSs need to manage unavoidable imprecise and uncertain information about constituent systems;
- Independence of constituents – constituent systems are independent entities for what concern operation, management, and evolution.

Each constituent has its own objectives, possibly conflicting with objectives of the other constituents or of the SoS. An SoS has a potentially limited authority that it exercises on the constituent systems to drive them towards its objectives.

An open problem of SoSs is how to provide justification of the reliance on emergent properties [22]. Emergent properties are the result of synergistic collaboration between constituent systems, and are those properties that cannot be expressed at the level of a single constituent system, but require observations of phenomena at the SoS boundary [22]. There are two different types of emergence: weak emergence [41], when an emergent property can be reduced to its constituents, and strong emergence, when an emergent property cannot be traced to any direct cause (a classic example is the consciousness emerging as a property of the brain [22]). Similarly to the approach in [53,22], we confine our attention to weak emergence, and hereafter, we use the term emergence to refer to weak emergence. Emergence can be either anticipated, meaning that it is defined at design-time, or unanticipated, meaning that it is not purposely or consciously designed-in or surprising to the developers and users of the SoS. Consequences of the unanticipated emergent behavior may be viewed as negative/harmful, positive/beneficial, or neutral/unimportant by stakeholders of the SoS [41]. Existing approaches to design and verify depend-

[27] https://www.ibm.com/blogs/internet-of-things/iot-design-for-system-of-systems.
[28] www.toyota-global.com/innovation/smart_mobility_society/intelligent_transport_systems.
[29] https://socrates2.org/.

able systems[30] work pretty well with closed and unchanging systems. These techniques become inadequate for assessing and justifying SoS reliance on emergent properties since SoSs are composed of autonomous constituents whose behavior can neither be predicted nor controlled [57]. Investigation is needed to provide ways for characterizing, quantifying and measuring a system behavior in the presence of perturbations [57].

One of the most frequently used techniques to explore emergence is simulation [17]. The work in [73] advocates that different engineering techniques are required at different scales, including software engineering and formal methods. Self-adaptive systems [16,14,51] are systems that autonomously decide how to adapt at runtime to meet environment (context) and user changes and threats. Uncertainty about constituents and independence of constituents make inapplicable for SoSs most of the techniques that have been proposed for self-adaptive system. Providing guarantees that an SoS will achieve a global goal in predictable and dependable ways, through the collaboration of autonomous and independent constituent systems with different (and potentially conflicting) local goals remains an open problem [18,34,75]. There is the need of evidence on which to base reliance on emergent properties and on SoS-level behaviors.

9. Conclusions and future work

In this paper we described our ongoing work in conceiving the next generation of electrical and software architecture of vehicles of the near future. The work is performed in the context of two Swedish projects led by Volvo Cars. In particular, in this work we focus on how to architect a car as a constituent system of a future transportation system. The outcome of this research is a viewpoint for vehicles that are constituents of SoSs, to be integrated into the Volvo Cars architecture framework proposed in [45]. For the new SoS viewpoint, we give the concerns that are framed based on our identified stockholders. The concerns are motivated by several scenarios that we have included in the paper. Also, we have included model kinds to be applied for the SoS viewpoint. We have demonstrated the value of the SoS viewpoint by applying it to a scenario where a car overtakes another car using information from different systems.

In the paper we also describe a functional reference architecture for vehicles that aim at being part of an SoS. As future work we plan to further validate the reference architecture and to provide an instantiation of it in the context of an OEM.

Moreover, as discussed in the paper, engineering efforts are needed also at the level of the system of systems as a whole. When constituent systems are developed by different organizations, which might have different owners, managers, and goals, other aspects need to be investigated. In general, there should be an engineering effort at the level of the whole SoS from the definition of the goals of the SoS, to the development, management, and maintenance of the constituent systems. Constituent systems should be able to join the SoS, to properly interact and interoperate with the other constituent systems, and to perform all the actions that are required to guarantee the achievement of the SoS goals, even at the cost of temporarily sacrificing the constituents' goals. Finally, the collaboration and interoperability of constituent systems in an SoS can potentially generate dangerous emergent behaviors that should be properly controlled and avoided.

Acknowledgements

This work was supported by the VINNOVA-projects NGEA and NGEA step 2. We would like to acknowledge all the partners of the project that made possible to achieve the results described in this paper. In particular, we would like to acknowledge the co-authors of the previous publication that are not contributing in this extended version, namely, Avenir Kobetski, Tony Larsson, Maytheewat Aramrattan, Tobias Aderum, Göran Jonsson, and Anders Thorsén. One of the authors (Patrizio Pelliccione) also acknowledges financial support from Centre of EXcellence on Connected, Geo-Localized and Cybersecure Vehicle (EX-Emerge), funded by Italian Government under CIPE resolution n. 70/2017 (Aug. 7, 2017).

References

[1] A. Yip, et al., Privacy-preserving browser-side scripting with BFlow, in: EuroSys, 2009.
[2] A. Avizienis, J.-C. Laprie, B. Randell, C. Landwehr, Basic concepts and taxonomy of dependable and secure computing, IEEE Trans. Dependable Secure Comput. 1 (1) (Jan 2004) 11–33.
[3] J. Axelsson, Business models and roles for mediating services in a truck platooning system-of-systems, in: 14th Annual Conference System of Systems Engineering, SoSE 2019, 19 May through 22 May, 2019, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 113–118.
[4] S. Behere, M. Törngren, A functional reference architecture for autonomous driving, Inf. Softw. Technol. 73 (May 2016) 136–150, Butterworth-Heinemann, Newton, MA, USA.
[5] D.E. Bell, L. La Padula, Secure Computer System: Unified Exposition and Multics Interpretation, MITRE Corporation, Bedford, MA, 1976, Tech. Rep. MTR-2997, Rev. 1.
[6] M. Benza, C. Bersani, M. D'Inca, C. Roncoli, R. Sacile, A. Trotta, D. Pizzorni, S. Briata, R. Ridolfi, Intelligent transport systems (its) applications on dangerous good transport on road in Italy, in: 2012 7th International Conference on System of Systems Engineering, SoSE, July 2012, pp. 223–228.
[7] C. Bergenhem, Approaches for facilities layer protocols for platooning, in: 2015 IEEE 18th International Conference on Intelligent Transportation Systems, ITSC, IEEE, 2015, pp. 1989–1994.
[8] C. Bergenhem, S. Shladover, E. Coelingh, C. Englund, S. Tsugawa, Overview of platooning systems, in: Proceedings of the 19th ITS World Congress, Oct. 22–26, Vienna, Austria, 2012.
[9] T. Bijlsma, T. Hendriks, A fail-operational truck platooning architecture, in: 2017 IEEE Intelligent Vehicles Symposium, IV, IEEE, 2017, pp. 1819–1826.

[30] Dependability is a generic concept including attributes such as reliability, availability, safety, integrity, maintainability, etc. [2].

[10] J. Boardman, B. Sauser, System of systems – the meaning of of, in: 2006 IEEE/SMC International Conference on System of Systems Engineering, April 2006, p. 6.
[11] F. Boeira, M.P. Barcellos, E.P. de Freitas, A. Vinel, M. Asplund, Effects of colluding Sybil nodes in message falsification attacks for vehicular platooning, in: 2017 IEEE Vehicular Networking Conference, VNC, Nov 2017, pp. 53–60.
[12] M. Broy, M. Gleirscher, S. Merenda, D. Wild, P. Kluge, W. Krenzer, Toward a holistic and standardized automotive architecture description, Computer 42 (12) (Dec 2009) 98–101.
[13] D. Burcicki, The criticality of the automotive E/E architecture, EENEWS Eur. Automot. (2019).
[14] B.H. Cheng, R. de Lemos, H. Giese, P. Inverardi, J. Magee, J. Andersson, B. Becker, N. Bencomo, Y. Brun, B. Cukic, et al., Software engineering for self-adaptive systems: a research roadmap, in: Software Engineering for Self-Adaptive Systems, Springer, 2009, pp. 1–26.
[15] J.S. Dahmann, K.J. Baldwin, Understanding the current state of us defense systems of systems and the implications for systems engineering, in: 2008 2nd Annual IEEE Systems Conference, April 2008, pp. 1–7.
[16] R. De Lemos, H. Giese, H.A. Müller, M. Shaw, J. Andersson, M. Litoiu, B. Schmerl, G. Tamura, N.M. Villegas, T. Vogel, et al., Software Engineering for Self-Adaptive Systems: A Second Research Roadmap, Springer, 2013.
[17] T. De Wolf, T. Holvoet, Towards a methodology for engineering self-organising emergent systems, in: Proceedings of the 2005 Conference on Self-Organization and Autonomic Informatics (I), IOS Press, 2005, pp. 18–34.
[18] N. Delgado, A.Q. Gates, S. Roach, A taxonomy and catalog of runtime software-fault monitoring tools, IEEE Trans. Softw. Eng. 30 (12) (Dec. 2004) 859–872.
[19] U. Eliasson, R. Heldal, P. Pelliccione, J. Lantz, Architecting in the automotive domain: descriptive vs prescriptive architecture, in: Proceedings of WICSA2015, May 2015, pp. 115–118.
[20] ETSI, En 302 637-3 v1.2.2 – Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications, Part 3: Specifications of Decentralized Environmental Notification Basic Service, 2012.
[21] ETSI, En 302 637-2 v1.3.2 – Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications, Part 2: Specification of Cooperative Awareness Basic Service, 2014.
[22] J. Fitzgerald, P.G. Larsen, J. Woodcock, Foundations for Model-Based Engineering of Systems of Systems, Springer International Publishing, Cham, 2014, pp. 1–19.
[23] H. Günther, B. Mennenga, O. Trauer, R. Riebl, L. Wolf, Realizing collective perception in a vehicle, in: 2016 IEEE Vehicular Networking Conference, VNC, Dec. 2016, pp. 1–8.
[24] R. Heldal, P. Pelliccione, U. Eliasson, J. Lantz, J. Derehag, J. Whittle, Descriptive vs prescriptive models in industry, in: Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems, MODELS '16, ACM, New York, NY, USA, 2016, pp. 216–226.
[25] International A. K., Global Automotive Executive Survey: From a Product-Centric World to a Service-Driven Digital Universe, 2017.
[26] International Organization for Standardization (ISO), 26262:2018 – Road vehicles – functional safety, 2018.
[27] ISO/IEC/IEEE, 24765:2017 – Systems and software engineering – vocabulary, 2017.
[28] ISO/IEC/IEEE, 42010:2011 – Systems and software engineering – architecture description, 2011.
[29] R.M. Jaradat, C.B. Keating, J.M. Bradley, A histogram analysis for system of systems, Int. J. Syst. Syst. Eng. 5 (3) (2014) 193–227.
[30] R. Johansson, J. Nilsson, C. Bergenhem, S. Behere, J. Tryggvesson, S. Ursing, A. Söderberg, M. Törngren, F. Warg, Functional safety and evolvable architectures for autonomy, in: Automated Driving, Springer, 2017, pp. 547–560.
[31] A. Knauss, J. Schroeder, C. Berger, H. Eriksson, Paving the roadway for safety of automated vehicles: an empirical study on testing challenges, in: Proceedings of Intelligent Vehicle Symposium, IV, 2017.
[32] M.N. Krohn, A. Yip, M.Z. Brodsky, N. Cliffer, M.F. Kaashoek, E. Kohler, R.T. Morris, Information flow control for standard OS abstractions, in: SOSP 2007, 2007.
[33] P. Langeland, R. Phillips, Heavy vehicles and traffic accidents – Norway versus other European countries, TØI-rapport, 2015.
[34] M. Leucker, C. Schallhart, A brief account of runtime verification, in: The 1st Workshop on Formal Languages and Analysis of Contract-Oriented Software, FLACOS07, J. Log. Algebraic Program. 78 (5) (2009) 293–303.
[35] N. Leveson, Engineering a Safer World: Systems Thinking Applied to Safety, Mit Press, 2011.
[36] K.-Y. Liang, J. Mårtensson, K.H. Johansson, Heavy-duty vehicle platoon formation for fuel efficiency, IEEE Trans. Intell. Transp. Syst. 17 (4) (2015) 1051–1061.
[37] N. Lyamin, D. Kleyko, Q. Delooz, A. Vinel, Real-time jamming dos detection in safety-critical V2V C-ITS using data mining, IEEE Commun. Lett. (2019) 442–445.
[38] M.W. Maier, Architecting principles for systems-of-systems, in: INCOSE International Symposium, Vol. 6, 1996, pp. 565–573.
[39] I. Malavolta, P. Lago, H. Muccini, P. Pelliccione, A. Tang, What industry needs from architectural languages: a survey, IEEE Trans. Softw. Eng. 39 (6) (June 2013) 869–891.
[40] A.C. Myers, L. Zheng, S. Zdancewic, S. Chong, N. Nystrom, Jif: Java Information Flow, http://www.cs.cornell.edu/jif, 2001.
[41] C.B. Nielsen, P.G. Larsen, J. Fitzgerald, J. Woodcock, J. Peleska, Systems of systems engineering: basic concepts, model-based techniques, and research directions, ACM Comput. Surv. 48 (2) (Sep. 2015), 18:1–18:41.
[42] J. Nilsson, C. Bergenhem, J. Jacobson, R. Johansson, J. Vinter, Functional safety for cooperative systems, SAE Technical Paper, SAE International, 2013.
[43] R.L. Nord, I. Ozkaya, P. Kruchten, Agile in distress: architecture to the rescue, in: International Conference on Agile Software Development, Springer, 2014, pp. 43–57.
[44] G. Panahandeh, E. Ek, N. Mohammadiha, Road friction estimation for connected vehicles using supervised machine learning, in: IEEE Intelligent Vehicles Symposium, IV, 2017.
[45] P. Pelliccione, E. Knauss, R. Heldal, S.M. Ågren, P. Mallozzi, A. Alminger, D. Borgentun, Automotive architecture framework: the experience of Volvo Cars, J. Syst. Archit. 77 (2017) 83–100.
[46] P. Pelliccione, A. Kobetski, T. Larsson, M. Aramrattan, T. Aderum, S.M. Ågren, G. Jonsson, R. Heldal, C. Bergenhem, A. Thorsén, Architecting cars as constituents of a system of systems, in: Software-Intensive Systems-of-Systems, ACM, 2017.
[47] J. Rasmussen, Risk management in a dynamic society: a modelling problem, Saf. Sci. 27 (2) (1997) 183–213.
[48] A. Russo, Functional pearl: two can keep a secret, if one of them uses Haskell, in: ICFP 2015, 2015.
[49] A. Sabelfeld, A.C. Myers, Language-based information-flow security, IEEE J. Sel. Areas Commun. 21 (1) (Jan. 2003) 5–19.
[50] SAE On-Road Automated Vehicle Standards Committee and Others, Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. SAE Standard J. 3016, 2018, pp. 1–16.
[51] M. Salehie, L. Tahvildari, Self-adaptive software: landscape and research challenges, ACM Trans. Auton. Adapt. Syst. 4 (2) (May 2009), 14:1–14:42.
[52] T. Samad, T. Parisini, Systems of systems, in: The Impact of Control Technology, 2011, pp. 175–183.
[53] J. Sanders, G. Smith, Emergence and Refinement, Form. Asp. Comput. 24 (2012) 45–65.
[54] D.C. Shoup, The High Cost of Free Parking, updated edition, American Planning Association Planners Press, Chicago, 2011.
[55] R. Spalazzese, P. Pelliccione, U. Eklund, INTERO: An Interoperability Model for Large Systems, IEEE Softw. (2017).
[56] D. Stefan, E.Z. Yang, P. Marchenko, A. Russo, D. Herman, B. Karp, D. Mazières, Protecting users by confining JavaScript with COWL, in: OSDI 14, 2014.

[57] L. Strigini, Fault tolerance and resilience: meanings, measures and assessment, in: Resilience Assessment and Evaluation of Computing Systems, 2012, pp. 3–24.
[58] A. Tolk, What comes after the semantic web – PADS implications for the dynamic web, in: Proceedings of the 20th Workshop on Principles of Advanced and Distributed Simulation, PADS '06, IEEE Computer Society, Washington, DC, USA, 2006.
[59] E. Uhlemann, The battle of technologies or the battle of business models? [connected vehicles], IEEE Veh. Technol. Mag. 13 (1) (March 2018) 14–18.
[60] R.A. Uzcategui, A.J.D. Sucre, G. Acosta-Marum, Wave: a tutorial, IEEE Commun. Mag. 47 (5) (May 2009) 126–133.
[61] P. van Staa, How KETSs can contribute to the re-industrialisation of Europe, in: European Technology Congress, Wroclaw, 2014, June 12-13, 2014, http://docplayer.net/21724658-Date-2012-how-kets-can-contribute-to-the-re-industrialisation-of-europe.html.
[62] A. Vinel, L. Lan, N. Lyamin, Vehicle-to-vehicle communication in C-ACC/platooning scenarios, IEEE Commun. Mag. 53 (8) (August 2015) 192–197.
[63] A. Vinel, N. Lyamin, P. Isachenkov, Modeling of V2V communications for C-ITS safety applications: a CPS perspective, IEEE Commun. Lett. 22 (8) (Aug. 2018) 1600–1603.
[64] E. Vinkhuyzen, M. Cefkin, Developing socially acceptable autonomous vehicles, in: Ethnographic Praxis in Industry Conference Proceedings, Vol. 2, 2016, pp. 522–534.
[65] S. Voget, SAFE RTP: an open source reference tool platform for the safety modeling and analysis, in: Embedded Real Time Software and Systems, ERTS2014, Toulouse, France, Feb. 2014.
[66] W. Wang, A. Tolk, W. Wang, The levels of conceptual interoperability model: applying systems engineering principles to m&s, in: Proceedings of the 2009 Spring Simulation Multiconference, SpringSim '09, Society for Computer Simulation International, San Diego, CA, USA, 2009, pp. 168:1–168:9.
[67] WHO, Global Status Report on Road Safety 2015, World Health Organization, 2015, ISBN 978 92 4 156506 6.
[68] R. Wohlrab, U. Eliasson, P. Pelliccione, R. Heldal, Improving the consistency and usefulness of architecture descriptions: guidelines for architects, in: Proceedings of the IEEE International Conference on Software Architecture, ICSA 2019, Hamburg, Germany, 2019.
[69] R. Wohlrab, P. Pelliccione, E. Knauss, R. Heldal, On interfaces to support agile architecting in automotive: an exploratory case study, in: Proceedings of the IEEE International Conference on Software Architecture, ICSA 2019, Hamburg, Germany, 2019.
[70] R. Wohlrab, P. Pelliccione, E. Knauss, M. Larsson, Boundary objects in agile practices: continuous management of systems engineering artifacts in the automotive domain, in: Proceedings of the 2018 International Conference on Software and System Process, ICSSP 2018, Gothenburg, Sweden, May 26-27, 2018, pp. 31–40.
[71] R. Wohlrab, P. Pelliccione, E. Knauss, M. Larsson, Boundary objects and their use in agile systems engineering, J. Softw. Evol. Process 31 (5) (2019) e2166.
[72] G.-r. You, X. Sun, M. Sun, J.-m. Wang, Y.-w. Chen, Bibliometric and social network analysis of the sos field, in: 2014 9th International Conference on System of Systems Engineering, SOSE, IEEE, 2014, pp. 13–18.
[73] F. Zambonelli, A. Omicini, Challenges and research directions in agent-oriented software engineering, Auton. Agents Multi-Agent Syst. 9 (3) (2004) 253–283.
[74] N. Zeldovich, S. Boyd-Wickizer, E. Kohler, D. Mazières, Making information flow explicit in HiStar, in: Proc. of the 7th USENIX Symp. on Operating Systems Design and Implementation, USENIX, 2006.
[75] P. Zhang, P. Pelliccione, H. Leung, X. Li, Automatic generation of predictive monitors from scenario-based specifications, Inf. Softw. Technol. 98 (2018) 5–31.
[76] G. Zyskind, O. Nathan, A. Pentland, Decentralizing privacy: using blockchain to protect personal data, in: 2015 IEEE Security and Privacy Workshops, May 2015, pp. 180–184.