Received November 2, 2019, accepted November 11, 2019, date of publication November 14, 2019, date of current version November 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2953568

Automotive Software in Connected and Autonomous Electric Vehicles: A Review

HRVOJE VDOVIC, (Graduate Student Member, IEEE), JURICA BABIC, (Member, IEEE), AND VEDRAN PODOBNIK, (Senior Member, IEEE)
Faculty of Electrical Engineering and Computing, University of Zagreb, 10000 Zagreb, Croatia

Corresponding author: Hrvoje Vdovic (hrvoje.vdovic@fer.hr)

This work was supported by the European Regional Development Fund under Grant KK.01.1.1.01.0009 (DATACROSS), Grant KK.01.2.1.01.0020 (RASCO-FER-SMART-EV), and Grant KK.01.2.1.01.0077 (bigEVdata).

ABSTRACT In the last decade, electric vehicles (EVs) have emerged as a sustainable transportation alternative to traditional internal combustion engine (ICE) cars, with automotive software as the key driver behind the advancements. A well-defined information and communication technology (ICT) architecture, comprised of electronics and software, can increase an EV's energy and cost efficiency, safety and comfort. With connected and autonomous electric vehicles (CAEVs) fast becoming a reality, the importance of software in vehicles increases tenfold. This paper serves as an introduction into the field of electromobility and automotive software. It provides an overview of the software and ICT architecture found in CAEVs and identifies future trends and challenges in automotive software development.

INDEX TERMS Automotive software, electric vehicles, connected vehicles, autonomous vehicles.

I. INTRODUCTION

In the last several decades, software has become an essential part of everyday life. Working, shopping, traveling, communicating and playing are all infused with and mediated by software [1]. Just as steam was the driver of the industrial age in the 19th century, software is the driver of today's information age. There is not a single industry today that is not at least partially governed by software, and automotive industry is no different.

In 2008, a conservative assessment of the average software value per car was \$425 and it is estimated to increase to \$575 by 2020 [2], indicating a rise in value of over 35%. Furthermore, the worldwide market value of automotive embedded software is estimated to grow from \$30 billion in 2008 to over \$52 billion in 2020 [2]. All of this, including the growing complexity of software in automotive systems, indicates that software engineering is an increasingly important discipline within automotive systems development and it will become even more crucial for vehicle manufacturers to equip their vehicles with software of the highest quality to stay relevant in today's market.

Software is an indispensable part of today's vehicles and its importance is perhaps the most significant in electric vehicles (EVs), as they are actively pushing for a bigger share in the automotive market. The number of electric cars surpassed 2 million vehicles worldwide in 2016 after crossing the 1 million threshold in 2015 [9]. Country targets, EV equipment manufacturer announcements and continuous technology improvements indicate that the electric car stock will range between 9 million and 20 million by 2020 and between 40 million and 70 million by 2025 [9]. With the advent of electric vehicles comes the increase of the number of interconnected electronic control units (ECUs) and, consequently, the complexity of software needed to manage them. Today, some vehicles contain upward of 100 ECUs, running millions of lines of software code [10].

It is estimated that by 2020 75% of cars will be built with the necessary hardware to connect to the Internet [11], enabling automated links to all other connected objects such as smart-phones, tracking devices, traffic lights and other motor vehicles [12]. The interaction with other devices and information systems creates an information-rich travel environment for the passengers and enhances the situational awareness of the vehicle [13]. Software plays a prominent part in the development of connected cars, because the information collected from other vehicles and the grid has to be processed and displayed to the driver as quickly as possible.

Connected vehicles provide a basis for the emergence of autonomous driving. Autonomous vehicles are vehicles with action and motion capabilities which do not require any sort of driver or teleoperation control [14]. They are set to be the next step in the automotive evolution, bringing change not only to automotive industry, but to the way transportation is

The associate editor coordinating the review of this manuscript and approving it for publication was Razi Iqbal.

TABLE 1. Literature survey results.

Search Terms				Search Results			
				Number of Papers Found		Most Cited Paper (Number of Citations)	
Automotive Software	Electric Vehicle(s)	Connected	Autonomous	Scopus	IEEE Xplore	Scopus	IEEE Xplore
✓				694	331	[3] (278)	[4] (121)
✓	✓			4	4	[5] (25)	[5] (15)
✓		✓		26	23	[6] (97)	[6] (87)
✓			✓	32	28	[7] (16)	[7] (18)
✓	✓	✓		0	0	-	-
✓		✓	✓	5	6	[7] (16)	[7] (18)
✓	✓		✓	0	1	-	[8] (9)
✓	✓	✓	✓	0	0	-	-

perceived. Today, autonomous cars are being developed by several manufacturers [15] and software is taking some of the critical functions (e.g. controlling the vehicle and analyzing the vehicle's surroundings), with a goal to eventually replace human drivers.

Connected and autonomous vehicles (CAVs) are currently among the most researched automotive technologies. With the growing environmental concerns and greenhouse gas reduction goals, the future CAVs are expected to be electric, indicating that a connected and autonomous electric vehicle (CAEV) should become a common feature on the road within the next decade [16]. To be competitive, CAEVs need to have a well-designed information and communication technology (ICT) architecture in the form of on-board electronics, together with associated automotive software, as this could improve a vehicles energy efficiency, driving comfort and passenger safety, all of which are significant aspects to consider when buying a new vehicle. In addition to the ICT architecture design, CAEV software development offers challenges in the form of many quality requirements that need to be satisfied. *Software reliability* is critical since CAEVs are software-driven (e.g. drive-by-wire systems) and any kind of software failure can put the lives of the passengers to risk [17]. Next, the software has to be *secure* to prevent unauthorized access to the in-vehicle network and with modern vehicles "knowing" more about their passengers than ever before, it needs to ensure *privacy protection* [18]. Furthermore, *usability* is particularly important because of its impact on safety, as the vehicle interface must provide access to information and controls without distracting the driver [19]. Modern vehicles also need to be *adaptable* to the needs of the driver, enabling higher personalization, and software and hardware updates.

Bearing in mind the ever-increasing relevance of CAEVs and the importance of automotive software in them, a literature survey of the field was made (Table 1). The four search terms related to the focus of the paper were chosen for the survey: *automotive software* as the core search term, and *electric vehicle* (or *vehicles*), *connected* and *autonomous* as optional terms. The search terms were applied to the title, the abstract and keywords of the papers in Scopus, Elsevier's database of peer-reviewed literature, and to the metadata of conference, journal and magazine papers, books and early access articles in IEEE Xplore, the IEEE research database. Although there were 694 and 331 papers found in Scopus and IEEE Xplore databases respectively for the term *automotive software*, there was only four of them found for the combination of *automotive software* and *electric vehicles*, and none were found for the combination of all four search terms, suggesting that an overview of the research area is needed. Therefore, this paper aims to provide a systematic summary of automotive software found in today's CAEVs. Such a summary, in addition to providing an introduction to the field, could also be useful for the identification of trends and challenges which could potentially be tackled in future work.

The remainder of the paper has the following structure. Section II describes the automotive software and ICT architecture found in CAEVs, with Section III mapping that architecture on today's most popular EVs. In Section IV, the future trends and challenges of automotive software development are described, while Section V concludes the paper.

II. CAEV ICT ARCHITECTURE

The vast majority of innovations in vehicles is driven by ICT in the form of electronics and software, and in case of CAEVs, ICT becomes the backbone of all relevant system functions. In the following, an overview of the software and ICT architecture found in CAEVs will be provided, with a special emphasis on vehicle components running automotive software. The vehicle architecture will be divided into

TABLE 2. Comparison of automotive bus systems.

Bus System	CAN	FlexRay	LIN	MOST
Type	event-triggered	time-triggered	subbus	multimedia
Application	ABS, battery management system, engine control	drive-by-wire systems	door locking, powered windows	entertainment, navigation
Bandwidth	up to 1 Mbit/s	up to 10 MBit/s	up to 20 kbit/s	up to 24.8 Mbit/s
Access Control	CMSA/CD	TDMA	Polling	TDM, CSMA/CA
Error protection	CRC, parity bits	CRC, bus guardian	checksum, parity bits	CRC, checksum, parity bits

four groups related to the research area of this paper: *electric*, *softwarized*, *connected* and *autonomous*.

A. ELECTRIC ARCHITECTURE

Although there are different types of EVs (e.g. hybrid EVs, plug-in hybrid EVs and solar-powered EVs), today's highway-capable EVs are mostly battery electric vehicles (BEVs). Their architecture is fairly simple; it consists of a battery pack, battery management system, electric motor and a motor controller.

Battery pack is used for storing electric energy and it is the most expensive component in EVs [20]. Battery pack consists of series-connected battery cells, and its capacity determines the driving range of the EV. To keep the battery cells within their safe operating range a *Battery Management System (BMS)* is required [21]. BMS monitors the state of charge of battery cells, their voltage levels, the current drawn from the cells and cell temperature.

The key component of an EV is the *electric motor*, as it is responsible for performing the vehicle's main function - driving. The electric motor can also be used as a generator of electricity through regenerative or dynamic braking. Between the electric motor and the battery is an inverter, which converts the DC source from the battery pack to AC current driving the motor. To ensure efficient and reliable operation of the electric motor, a *motor controller* is needed. The controller normally controls the power supplied to the motor, and with it, the vehicle speed.

Finally, to create an energy efficient EV, it is not enough to simply replace the ICE and fuel tank in traditional ICE cars with an electric motor and battery; the entire powertrain has to be modelled for the motor and battery.

B. SOFTWARIZED ARCHITECTURE

In EVs, automotive software is the foundation of all vehicle functions. It would be impossible to count all the components running automotive software in EVs, so a high-level approach will be taken, providing an overview of automotive electronics, bus systems and software architecture. Well-defined software architecture is especially important in automotive software design, as it helps to understand and predict the performance of the system by examining all of its subsystems, components and modules. Software and electronics in automotive systems go hand in hand, which is why software architecture cannot be viewed as stand-alone, without being connected to vehicle electronic systems.

Electronic Control Units (ECUs) are embedded systems on which software that controls one or more electronic systems or subsystems in a vehicle is executed. Each ECU is responsible for performing a certain vehicle function. Assembly language or the C programming language can be used to program the ECUs [22], which, along with the software running on them, form the electrical and software system of the car. When electronics was first started being deployed in vehicles, the operation of ECUs was largely autonomous with no interaction among different ECUs.

The networking of different ECUs was made possible with the introduction of *automotive bus systems*. Automotive bus systems interconnect components inside a vehicle, allowing multiple access to various sensors and enabling higher-level software functions that run on several subsystems. They can be divided into four different communication groups based on their application area and technical properties:

- event-triggered bus systems (e.g. CAN);
- time-triggered bus systems (e.g. FlexRay);
- subnetworks (e.g. LIN);
- multimedia bus systems (e.g. MOST).

The comparison of the mentioned representatives of each automotive bus system communication group can be seen in Table 2.

Control Area Network (CAN) is an event-triggered bus system mainly used for soft real-time in-car serial communication between ECUs, networking for example the antilock braking system (ABS) [23] or the battery management system [24]. It enables data rates up to 1 Mbit/s and broadcasts messages to all connected nodes which independently decide whether to process the message or not. CAN ensures the quickest transmission of top priority messages using CSMA/CD (Carrier Sense Multiple Access / Collision Detection) access control method and it detects transfer errors using cyclic redundancy check (CRC) and parity bits, and initiates retransmission of affected messages [23].

FlexRay is a time-triggered bus system developed in 1999 for future safety-relevant high-speed automotive networks, targeting highly safety relevant applications such as drive-by-wire and powertrain [23]. It supports data rates up to 10 Mbit/s and up to 64 connected nodes. It uses the cyclic TDMA (Time Division Multiple Access) that provides fixed time slots during which devices can access the bus and mini-slots which can enlarge if devices have additional data to be sent that doesn't fit in their designated time slot [25]. The error tolerance is achieved by redundant channels, CRC and a bus guardian that detects and handles logical errors [23].

Local Interconnect Network (LIN) is a local sub network usually connected as a sub-bus to a more powerful bus, such as CAN. It has been developed for less complex networks where the bandwidth and versatility of CAN bus is not required, such as, automatic door locking mechanisms, power-windows and temperature and rain sensors [23]. LIN supports data transmission of up to 20 kbit/s and error protection in the form of checksums and parity bits.

Multimedia bus systems such as *Media Oriented Systems Transport (MOST)* are responsible for the transmission of high-quality audio, voice and video data streams used for in-vehicle entertainment. MOST features a data rate of up to 24.8 Mbit/s in synchronous and 14.4 Mbit/s in asynchronous transmission mode with a 700 kbit/s control channel [25]. Access control in synchronous mode is realized by TDM (Time Division Multiplex) and by CSMA/CA in asynchronous mode. An internal MOST system service is in charge of error management, detecting errors over parity bits, CRC and checksums.

Vehicle electronics and software architecture can be viewed as multiple subsystems comprised of different ECUs that communicate over bus systems. Although each subsystem is responsible for a certain part of the vehicle, they communicate with one another, distributing some of the more complex functions over several subsystems and ECUs. The main subsystems found in vehicles are:

- powertrain subsystem;
- chassis subsystem;
- body subsystem;
- multimedia subsystem.

The *powertrain subsystem* is in charge of the components that generate power and deliver it to the road surface; e.g. electric motor, motor controller, battery pack and battery management system. It has a relatively small number of user interfaces; the start/stop switch for starting and shutting off the vehicle and the accelerator pedal for increasing the speed of the vehicle. Along with controlling the powertrain according to user inputs, software functions of the powertrain subsystem include collecting information about the working parameters of the electric motor and battery pack, and notifying the driver via information systems if anything is out of order. The throttle-by-wire system, needed for the control of autonomous vehicles without human input, is also implemented in the powertrain subsystem of the electric vehicle.

The *chassis subsystem* encompasses vehicle's axles and wheels, brakes, steering system, suspension and shock absorbers. It includes systems such as the antilock braking system (ABS), electronic stability program (ESP), parking brake, tire pressure monitoring system, suspension, power steering, brake-by-wire and steer-by-wire systems [26]. Functional failure of any of the aforementioned systems can put the passengers' lives at risk, which is why the safety requirements for the chassis subsystem are very stringent, with sensitive monitoring mechanisms in place for each of the safety critical systems. Similar to the powertrain system, the chassis subsystem also has a small number of user interfaces; the brake pedal, the steering wheel and the parking brake.

The vehicle software and electronics responsible for passive safety and comfort and convenience are included in the *body subsystem* [26]. The *passive safety systems* are all onboard systems in charge of protecting the safety of vehicle's passengers (e.g. airbags, seat belt sensors and tighteners). The *comfort and convenience systems* include the central locking system, powered windows, wipers, heating and air conditioning, mirror adjusters, headlamp controls, parking aid features and other similar auxiliary systems which improve the passengers' travel comfort. Practically all of the comfort and convenience systems provide some kind of user interface, while passive safety systems usually provide none. The body subsystem contains the largest number of independent software functions due to the amount of different functionalities implemented in the subsystem.

The *multimedia subsystem* includes the instrument cluster and the infotainment system which provides added comfort value to the passengers. The instrument cluster usually displays vehicle information such as current speed, battery level and indicators about vehicle status, and it must comply with regulations and laws [27] to ensure that the information is properly displayed to the driver. To be able to display this information and to provide controls to elements from other subsystems, the ECUs of the multimedia subsystem are networked with the ECUs from the powertrain, chassis and body subsystems [26]. Infotainment system providing audio and video entertainment, voice control and smart phone integration is also a part of the multimedia subsystem. Examples of modern vehicle infotainment platforms are the QNX Car platform by Blackberry, Windows Embedded Automotive by Microsoft, GENIVI by GENIVI Alliance, Tizen IVI and Automotive Grade Linux by the Linux Foundation, and Android by Google [28].

Currently, most suppliers provide new functionalities to vehicles as hardware devices. This is one of the main reasons for the growing number of ECUs and the increase in complexity of the aforementioned subsystems. This approach is not scalable, so a transition from distributed hardware to distributed software and services will be necessary. For this purpose, flexible architectures and standardized software component approaches for vehicle software like AUTOSAR (Automotive Open System Architecture) will become essential for CAEVs [29].

AUTOSAR is a global standardization consortium of automotive manufacturers and suppliers developing an open industry standard for automotive electronic architectures. It addresses the challenge of rising code complexity by providing an open automotive software architecture which supports the development of standardized electronic systems that improve quality, performance, safety and environmental friendliness [30]. AUTOSAR enables modeling ECUs and entire electronic systems at a higher level using a top-down approach with the architecture of the ECU software as the starting point [31] and simplifies the process of updating software over the vehicle's lifetime.

AUTOSAR offers two solutions for the development of automotive software. AUTOSAR Classic Platform is the standard software platform for deeply embedded ECUs, and in CAEVs it remains the first choice for the development of software for the powertrain, chassis and body subsystems as they have high demands on safety, real-time and determinism while running on low cost hardware [7]. New use cases, such as connected and autonomous vehicles, required the development of the AUTOSAR Adaptive Platform. Adaptive platform is meant to be used in applications that demand high-end computing power, such as multimedia, automated driving and communication with traffic infrastructure and cloud servers.

C. CONNECTED ARCHITECTURE

Connected vehicle is a vehicle capable of communicating with other vehicles, the infrastructure, smart devices and, most importantly, the Internet. A connected vehicle can exchange information by several different types of communication - vehicle to vehicle, vehicle to road infrastructure, vehicle to internet and vehicle to smart device.

Vehicle To Vehicle (V2V) communication is the direct transmission of data between two or more vehicles. V2V communication can be used for accident avoidance, route optimization, multimedia information sharing and social interaction [28]. Communicating vehicles form a vehicular ad hoc network (VANET) which changes constantly and quickly. VANET enables communication between moving vehicles using dedicated short range communication (DSRC) [32], a short-range wireless communication channel designed for automotive use, standardized by the IEEE 802.11p standard, also known as WAVE (Wireless Access in Vehicular Environments) [33].

The WAVE system is also used in the *Vehicle To Road Infrastructure (V2I)* communication. The main system components enabling V2I communication are onboard units (OBUs) and roadside units (RSUs). An OBU is a device mounted on a vehicle used for exchanging information with RSUs (or with other OBUs in case of V2V communication) [32]. The RSUs are usually fixed along the side of the road or on light poles, traffic lights and road signs. RSU's main functions are extending the range of VANET by redistributing information from OBUs to other OBUs, running safety applications such as accident or work zone warnings, and providing Internet connectivity to OBUs [32].

Internet communication is the fundamental requirement for a connected vehicle. There are three different solutions by which a vehicle can achieve an Internet connection: an embedded solution, a tethered solution and an integrated solution [28]. In the embedded solution, a SIM card is permanently installed in the vehicle during the manufacturing process. The module carrying the SIM card in the embedded solution is called the Telematics Control Unit (TCU). The tethered solution offers a modem that allows the insertion of a personal SIM decoupled from the vehicle, while keeping the application intelligence inside the vehicle. In the integrated approach, both intelligence and Internet connectivity are provided by a user's smart device, with the applications running on the smartphone displayed on the vehicles interface.

Today, vehicle-to-Internet connection is most commonly achieved through 4G mobile communication. Connected vehicles generate high amounts of data, and with more and more vehicles becoming connected every year, the amount of generated data is set to increase. Vehicle manufacturers and telecommunications companies such as Audi, BWM, Deutsche Telekom, Ericsson and Vodafone have already started to prepare for the increase in vehicle generated data by forming the Fifth-Generation Automotive Association (5GAA) [34]. 5G mobile networks will have reduced latency, massive device connectivity and will be able to handle much more data, and the 5GAA aims to use this technology to address society's connected mobility needs.

Finally, connected vehicles need to offer a way to communicate with the user's personal smart devices. The *smart device integration* is usually achieved by a Bluetooth or Wi-Fi connection to the infotainment system. The infotainment system provides a user interface (UI) and the radio/entertainment/media functionality [35]. Current trends in infotainment systems steer away from embedded applications capable of running on only one system, and towards multiplatform applications capable of running on systems adopted by more than just one car manufacturer. The most widespread solutions for smart device integration are MirrorLink by Car Connectivity Consortium, AppLink by Ford, Apple CarPlay by Apple and Android Auto by the Open Automotive Alliance [28].

D. AUTONOMOUS ARCHITECTURE

An autonomous or a self-driving vehicle is a vehicle capable of perceiving its surroundings and navigating itself without human intervention. For autonomous driving to be made possible, a vehicle must be equipped with electronic control systems and surrounding detection sensors, along with complex autonomous driving algorithms, including:

- perception - sensing the surrounding environment using various types of sensor techniques such as RADAR, LIDAR and computer vision;
- localization - finding the car's position using the Global Positioning System (GPS);
- planning - determining the behaviour and motion of the vehicle based on information from perception and localization;
- control - executing the planned function by steering, accelerating and braking;
- system management - supervising the driving system with functions such as fault management and logging [36].

The information on surrounding environment is collected using several types of sensor components. *RADAR* system uses radio waves to determine the range, angle and velocity of objects. The *LIDAR* system creates a digital 3D representation of the surrounding objects by illuminating them with pulsed laser light and measuring the reflected pulses with a sensor. *Cameras* and *computer vision* are then used to recognize and distinguish various visual objects [37]. Localization is achieved by a *GPS system* using additional information such as vehicle motion sensors, environment perception data and digital maps to account for the errors that can occur in bad GPS satellite signal conditions. Planning algorithms are usually embedded in the vehicle's autopilot functionality and achieved entirely by software functions. Autopilot algorithms are divided in three stages: global routing (i.e. finding the fastest and safest way from the initial position to the goal position), behaviour reasoning (i.e. assessing the driving situation and determining the behaviour of the autonomous vehicle based on the global route and perception information), and local motion planning (i.e. avoiding static and dynamic obstacle collision) [37]. Autopilot software in autonomous vehicles also has some interesting *ethical requirements* as it needs to replicate the human decision making process [38].

The key technology enabling the control of an autonomous vehicle is *drive-by-wire*. Drive-by-wire is referring to the replacement of mechanical or hydraulic systems, such as acceleration (throttle-by-wire), braking (brake-by-wire) or steering (steer-by-wire), by electronic ones [39]. Additional ECUs are added so that the software in autonomous vehicles can perform all driving functions. Although drive-by-wire is emerging for ICE vehicles, it is especially important for EVs. In EVs, brake-by-wire technology enables energy saving in the form of energy recuperation during braking by decoupling the braking pedal from the brakes [20]. Naturally, drive-by-wire is a highly safety-critical system which is why it must be designed in a fault-tolerant fashion, with control system redundancies. For a critical drive-by-wire system, a management system must ensure that a system failure does not lead to endangerment of human lives or the vehicle environment, and that a failure of a single component does not lead to the failure of the whole drive-by-wire system [39].

E. HOLISTIC OVERVIEW OF THE CAEV ICT ARCHITECTURE

After a detailed analysis of each of the four architecture groups, it is useful to take a look at the ICT architecture as a whole to understand how various components interact with one another and how they fit in the complete system.

The complete CAEV ICT architecture can be seen on Figure 1. All four groups and components within them are interconnected and exchange information via automotive bus systems. In the *electric architecture* group, the battery pack comes paired with the battery management system while the electric motor comes paired with the motor controller. Battery management system and motor controller serve as interfaces toward the rest of the system, sending information about the battery and the motor, and controlling them according to received instructions. Subsystems inside the *sofwarized architecture* group receive user input from the driver and the passengers, and control the components which are under their responsibility in line with the user input. The infotainment systems found in CAEVs are dependent on the communication components installed in the vehicle which are a part of the *connected architecture* group. The *connected architecture* group is responsible for communication with the Internet, smart devices, other vehicles and the infrastructure. V2V and V2I technology is closely connected to the planning systems inside of the *autonomous architecture* group that uses the input from the surrounding vehicles, infrastructure, on-board cameras and sensors to determine vehicle behaviour. After a suitable behaviour is calculated, the vehicle is controlled via drive-by-wire systems found in the powertrain and chassis subsystems of the *softwarized architecture* group.

The ICT architecture of CAEVs is a complex system in which components must work in unison, but at the same time be independent enough to not allow the entire system to fail if a single component fails. Not to compromise the vehicle's safety, the historical evolutionary development of vehicle architecture avoided dramatic changes for as long as possible. Today's CAEV manufacturers are leaving legacy concepts behind and taking a holistic approach for handling all major activities in vehicle development.

III. ICT ARCHITECTURE OF TODAY'S MOST POPULAR EVs

Pure electric cars are steadily increasing their share on the car market. The five best selling vehicles of 2018 in the US were the Nissan Leaf with 14,715 units sold, Chevrolet Bolt with 18,019 units sold, Tesla Model S with 25,745 units sold, Tesla Model X with 26,100 units sold and Tesla Model 3 with 139,782 units sold [40] (for comparison, the overall top selling vehicle of 2018 in the US is the Ford F-Series with 909,330 units sold [41]). This section will provide an overview on the technology and software features provided in the Nissan Leaf, Chevrolet Bolt, and the most advanced of the three top selling Tesla models, the Tesla Model S (side-by-side comparison of the three EVs can be seen in Table 3).

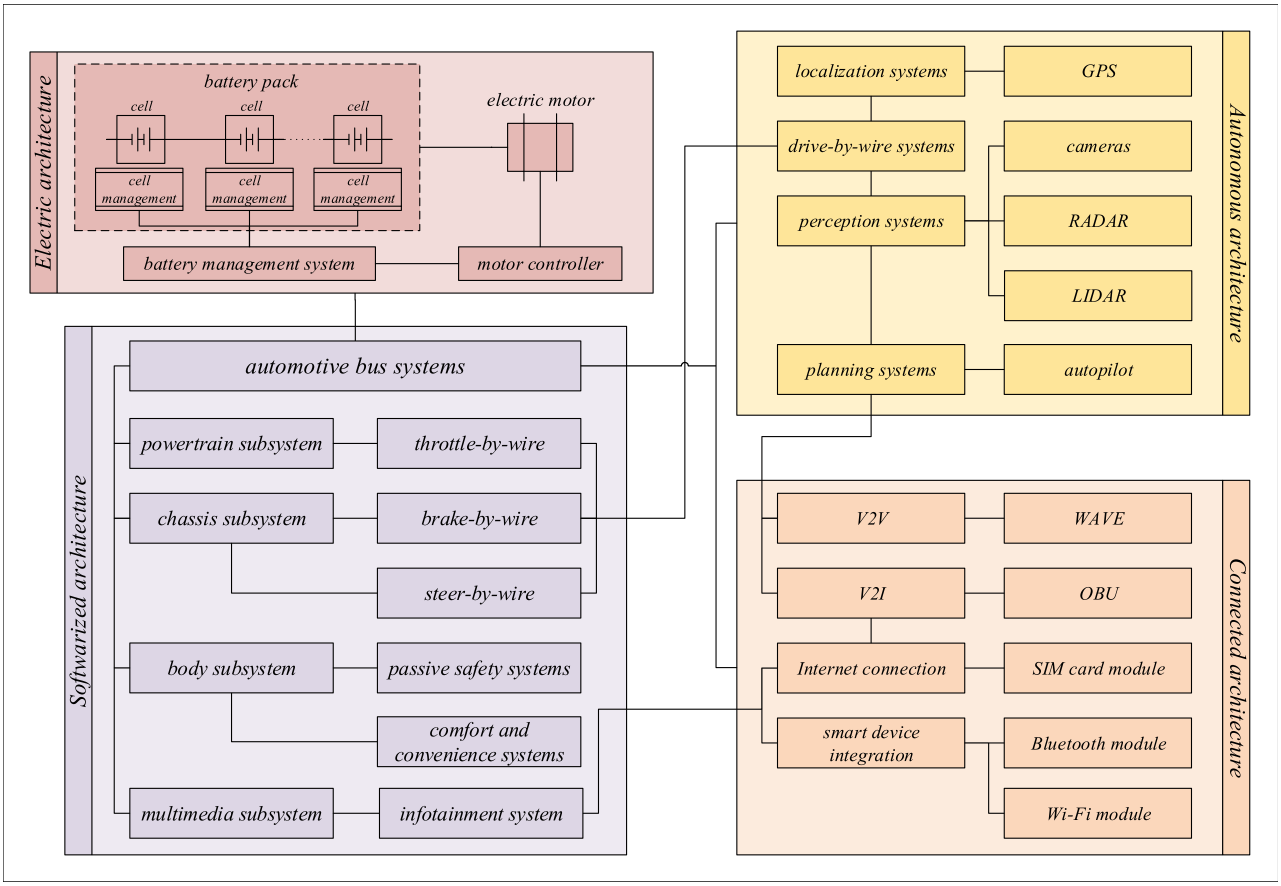


FIGURE 1. CAEV ICT architecture.

A. NISSAN LEAF

Nissan Leaf is a BEV manufactured by Nissan and introduced in 2010. The 2018 model is powered by a 40 kWh battery with the range of 243 km on a full battery charge, as measured by the U.S. Environmental Protection Agency (EPA) [42]. It has an AC synchronous electric motor with the maximum power of 110 kW capable of going up to 144 km/h and accelerating from 0 to 100 km/h in 7.9 seconds.

The number of ECUs managing the Nissan Leaf is between 3 and 8, depending on the model [43]. To enable the communication between different ECUs and various vehicle sensors and devices, different bus systems are installed on-board, including multiple CAN and LIN buses [44] and a FlexRay bus [45]. It has a half analogue, half digital instrument cluster. The speedometer is an analogue dial, with a digital vehicle information display next to it showing information about range, energy consumption and battery level. A seven-inch diagonal touchscreen is used to display navigation directions and information about vehicle surroundings. The operating system running on the touchscreen display is Windows Embedded Automotive, offering wired smart phone connection support with Android Auto and Apple CarPlay.

The Nissan Leaf connects to the Internet via an embedded SIM card used exclusively to access the vehicle with the NissanConnect EV application. The NissanConnect EV application offers functionalities such as checking the charge level and estimated driving range, and turning the battery charging or climate controls on or off [46]. V2V and V2I technology is currently not available in the Nissan Leaf.

Partial autonomous driving is available in the Nissan Leaf through its autopilot called Nissan ProPILOT. Nissan ProPILOT offers features such as Intelligent Emergency Braking, Intelligent Cruise Control, Intelligent Lane Intervention and automatic parking [47]. Intelligent Cruise Control maintains the distance to the car ahead, and Intelligent Emergency Braking applies brakes if distance gets too low to help avoid a collision. Intelligent Lane Intervention applies brakes to guide the car back into the lane. Nissan ProPILOT Park enables parking with the press of a button, controlling the vehicle instead of the driver. This is possible using the drive-by-wire technology and environment detection. Environment detection is achieved using two different camera systems, the RearView Monitor and the Around View Monitor [48]. RearView Monitor is a single camera system installed in the back of the car to give a view of what is behind it. Around View Monitor uses four wide-angle cameras mounted on the front, on each side mirror and on the rear to create a virtual composite 360° bird's-eye view of the vehicle. Nissan expects

TABLE 3. Comparison of most popular EVs over four research properties: electric, softwarized, connected and autonomous.

		2018 Nissan Leaf	2018 Chevrolet Bolt	Tesla Model S P100D
Price		$29,990	$36,620	$135,000
Electric	**Battery Capacity**	40 kWh	60 kWh	100 kWh
	Range	243 km	383 km	507 km
	Electric Motor Type	AC synchronous	permanent magnet synchronous	AC induction
	Max Engine Power	110 kW	150 kW	568 kW
	Top Speed	144 km/h	145 km/h	249 km/h
	Acceleration (0 - 100 km/h)	7.9 s	6.7 s	3.8
Softwarized	**Number of ECUs**	3 - 8	3 - 8	3 - 4
	Bus Systems	multiple CAN buses, multiple LIN buses, FlexRay	multiple CAN buses, multiple LIN buses, FlexRay	multiple CAN buses, multiple LIN buses, Ethernet
	Infotainment Panel	vehicle information display, 7-inch touchscreen	8-inch driver information center, 10.2-inch touchscreen	driver display, 17-inch touchscreen
	Infotainment Operating System	Windows Embedded Automotive, support for Android Auto and Apple CarPlay	MyLink based on QNX OS, support for Android Auto and Apple CarPlay	modified version of Ubuntu Linux
Connected	**V2V**	N/A	N/A	N/A
	V2I	N/A	N/A	N/A
	Internet Communication	embedded SIM card	embedded SIM card	embedded SIM card
	Smart Device Integration	NissanConnect EV application	built-in Wi-Fi hotspot, myChevrolet Mobile App	Tesla application
Autonomous	**Perception**	RearView Monitor, Around View Monitor	ultrasonic-, radar- and camera-based systems	8 cameras, 12 long range ultrasonic sensors, front radar
	Localization	GPS navigation included	GPS navigation included	GPS navigation included
	Drive-by-wire	fully drive-by-wire	fully drive-by-wire	fully drive-by-wire
	Autopilot	Nissan ProPILOT	N/A	Tesla Autopilot
	Autopilot Features	Intelligent Emergency Braking, Intelligent Cruise Control, Intelligent Lane Intervention, Automatic parking	N/A	Adaptive Cruise Control, Autopark/Summon, Autosteer, Speed assist
	Fully Autonomous Projection	2020	2020	2020

to introduce a fully autonomous Nissan Leaf to the production system by 2020 [49].

B. CHEVROLET BOLT

Chevrolet Bolt (also known as Opel Ampera-e in Europe) is a BEV developed and manufactured by Chevrolet in partnership with LG Corporation. The 2018 model has the EPA measured maximum range of 383 km [50] mainly because of its 60 kWh battery capacity. Chevrolet Volt is equipped with a 150 kW permanent magnet synchronous electric motor enabling the top speed of 145 km/h and capable of accelerating from 0 to 100 km/h in 6.7 seconds [51].

Like the Nissan Leaf, the Chevrolet Bolt can also have between 3 and 8 ECUs [43], interconnected with multiple CAN and LIN buses, along with a FlexRay bus enabling the drive-by-wire technology. The infotainment system in the

Chevrolet Bolt consists of two screens: an 8-inch diagonal driver cluster and a 10.2-inch diagonal color touchscreen. The 8-inch display is used as a driver information center providing detailed vehicle information with the option of customizing the type of data displayed [52]. Navigation directions, vehicle surroundings and media can be displayed on the 10.2-inch touchscreen. The infotainment system runs an operating system called MyLink, which is based on QNX, a Unix-like real-time operating system. Apple CarPlay and Android Auto compatibility is included in the OS and the features they offer can be used by connecting a compatible smart phone to the available USB data port.

Chevrolet Bolt is equipped with an embedded 4G LTE capable SIM card. The vehicle's internet connection can be shared with up to seven compatible devices using it's built in Wi-Fi hotspot [52]. The myChevrolet Mobile App, available for iOS and Android, gives the ability to control the vehicle with a mobile device. Some of the features the application offers are remote starting and stopping the engine, locking and unlocking the vehicle, locating the vehicle and warming the inside of the car while it is charging [53]. V2V and V2I technology is not available in the Chevrolet Bolt.

A number of ultrasonic-, radar- and camera-based systems are installed in the Chevrolet Bolt to improve vehicle safety. The rear vision camera captures the area behind the vehicle and the surround vision camera system creates a virtual bird's-eye view of the vehicle, both of which can be seen on the central touchscreen. Bolt does not offer an autopilot system, but some of the semi-autonomous features such as forward collision alert with automatic braking, lane keep assist with lane departure warning, and lane change blind spot alert [52] are made available using the perception systems and drive-by-wire technology. Fully autonomous Chevrolet Bolt, called the Cruise AV is expected to launch in 2020 at earliest [54].

C. TESLA MODEL S

Tesla Model S is a luxury BEV introduced in 2012 and manufactured by Tesla, Inc. Current top-of-the-line version of the Model S, the P100D, has a 100 kWh battery and the EPA measured maximum range of 507 km [55]. It has an AC induction electric motor with maximum power of 568 kW capable of going from 0 to 100 km/h in 3.8 seconds with the top speed of 249 km/h.

Depending on the version, the Tesla Model S can have 3 or 4 ECUs [43] controlling different vehicle subsystems. Multiple CAN and LIN buses along with high-speed Ethernet [56] are used to connect vehicle sensors and devices to the appropriate ECUs. The Tesla Model S has two digital displays: a driver display located behind the steering wheel and a central 17-inch touchscreen. The driver display shows information about current speed, energy consumption, available range and is customizable with options to display information about navigation, media playback and ongoing phone conversation if a smart phone is connected to the vehicle. The 17-inch central touchscreen is operated by a modified version of the Ubuntu OS with Tesla custom software and apps built on top of it, running everything from media, maps and calendars to a web browser [57].

Tesla Model S has cellular Internet connectivity included via an embedded SIM card. Tesla offers their smart phone application which can connect to the vehicle through the cellular network. It offers features such as unlocking and driving the vehicle without the key, checking the range and battery charge status, controlling the climate, locating the vehicle and the ability to summon the vehicle; making it drive forwards or backwards autonomously, avoiding obstacles [58]. Tesla currently does not incorporate the V2V and V2I technology into their vehicles.

Tesla Autopilot is an advanced driver-assistance system made by Tesla. It uses the vehicle installed perception systems including 8 cameras, 12 long range ultrasonic sensors and a front radar, which enable 360 degrees of vision around the car at up to 250 meters of range. Data processing is made by the Tesla-developed neural net for vision, sonar and radar processing, which construct the car's environment in a great level of reliability [59]. The Tesla Autopilot can match speed to traffic conditions, keep within a lane, change lanes automatically, exit the freeway when the destination is near, and transition from one freeway to another. It can also self-park when a parking spot is near and be summoned to and from a garage. All vehicles produced by Tesla have the hardware needed for full self-driving capability and Tesla estimates that by the middle of 2020, their autonomous system will have improved to the point where drivers will not have to pay attention to the road [60].

IV. FUTURE TRENDS AND CHALLENGES

Over the past 40 years, software has made significant innovations in the automotive domain possible. Today's automotive software engineering is spearheading IT innovation; encompassing modern embedded and cloud technologies, distributed computing, real-time systems, mixed safety and security systems and the connection of it all to long-term sustainable business models [61]. When looking at the importance of software for the modern consumer, a parallel can be drawn between personal computers, mobile phones and automobiles. In all three cases, consumers used to focus solely on the hardware aspect of the product: be it the processing power, graphics capabilities and available memory in PCs; the screen size, storage space and network capabilities in mobile phones; or the available horsepower, speed and acceleration in cars. Today, consumers tend to focus equally, if not more, on the software aspect of the product; looking at the user experience and functionalities provided by the supplied software, making it one of the key differentiating factors when purchasing a product. With the future of CAEVs relying on software, it is important to identify the emerging challenges and application constraints as soon as possible. In order to do that, a knowledge of new trends in the rapidly changing automotive industry is required.

A. TRENDS IN AUTOMOTIVE SOFTWARE AND SYSTEM DESIGN

The trends in automotive software and ICT architecture design are driven by two streams: societal and technological. The societal trends approach the future of mobility from an environmental point of view and the point of view of vehicle passengers. On the other hand, the technological trends back the societal requirements, while steering the automotive systems development in a way that incorporates emerging technologies and future sustainability.

According to Buckl *et al.* [62], the main societal trends shaping the automotive software development today are:

1) *Energy and cost efficiency:* The required energy per kilometer needs to be reduced. This can be achieved by replacing existing hardware functions with software, which decreases the vehicle's weight, or by adding intelligent predictive management functions.
2) *Zero accidents:* To reduce the number of accidents on the road, a number of pro-active safety functions need to be implemented. Fully autonomous vehicles are expected to be the ultimate solution for reducing the number of accidents.
3) *Seamless connectivity:* Modern vehicles will need to enable connections to other smart devices and the cloud, and provide frequent software updates to keep track with advances in multimedia and infotainment technology.
4) *Personalization:* Closely connected to the demand for seamless connectivity is the demand for personalization. Certain secondary vehicle functions will have to be transferred to a personal mobile device to provide users with information about their vehicle wherever they go. Vendor independent infotainment systems could allow users to create a consistently similar driving experience in different cars by a simple transfer of predefined settings from a mobile device.

With the constantly changing landscape of the automotive industry, predicting and leveraging rising technologies is of paramount importance to the automotive software developers. Based on the CAEV ICT architecture and the ICT architectures of today's most popular EVs described in this paper, the main technological trends in today's automotive software industry can be identified:

1) *Computer architecture centralization:* The large number of ECUs in vehicles is causing the ICT architecture in vehicles to become more complex than required, increasing the needed integration and testing effort to exclude undesired interaction when adding new vehicle functionalities. Future architectures should be based on centralized and scalable computing units, with standardized hardware reducing the maintenance costs [62].
2) *Standardized communication:* Vehicles currently use several standards for in-vehicle communication. For the next generation vehicles and particularly EVs, Ethernet is expected to be the replacement for current communication buses because of its high-speed, cost efficient and light-weight implementation [20].
3) *Connectivity and cooperation:* Cars will have to be able to communicate with each other and the infrastructure through mobile and ad-hoc networks [63]. All modern cars are expected to provide smart device integration through Bluetooth or Wi-Fi connectivity.
4) *Autonomous functions:* Creating self-driving vehicles is the next step in automotive evolution and it entails a large amount of complexity in various vehicle systems [64]. Safety-critical systems, as well as verification and validation methods for software in cars will have to become even more stringent and advanced [61].

B. CHALLENGES

Automotive software, as a dominant factor in the automotive industry, brings various challenges whose solutions can prove decisive for vehicle manufacturers to stay ahead of the competition. Future trends can be used to identify the main challenges in the automotive software and ICT architecture design that the manufacturers have to solve. These challenges are:

1) *Reducing ICT architecture complexity:* The evolution of vehicle ICT architectures shows a trend of architectures becoming more complex than required for the corresponding gain in functionalities. This trend is identified for some time, and vehicle manufacturers have already moved to answer it. The number of ECUs in an average vehicle in 2010 was around 70 [62], while in today's electric vehicles it goes as low as three (as seen on Table 3). The centralization of ECUs impacts a vehicle's weight and cost, because of the decreased number of needed components, wiring optimizations, and reduced integration and testing effort. The ECU consolidation is almost an imperative in autonomous vehicles, as centralization decreases the complexity of development of advanced assisted driving functions, reduces the number of operations that can go wrong and increases reliability.The hardware architecture of such a centralized system should consist of a small number of central processing units parallelized to ensure redundancy, interconnected with smart sensors and actuators via a homogeneous and optionally redundant network [65]. Perceived events, like video camera signals can be evaluated much faster if the processing of signal input, decision and output is located in a single system [66]. The relocation of software responsibility on a single, central ECU indicates that the workload on the ECU will increase, so the developed automotive software needs to be extremely resource efficient to decrease the hardware cost.

   Another way to reduce vehicle ICT architecture complexity is to use a domain controlled architecture, which reorders the corresponding vehicle functionality in a domain specific manner (typical domains are described in Section II-B powertrain, chassis, body and multimedia) [66]. In this architecture model, every domain has an additional abstraction layer in the form of server ECUs called domain controllers. Every domain controller can control several bus systems, deactivating them individually to save energy when their operation is not needed. Furthermore, applications which must always be available during car usage can be integrated on domain controller ECUs and can be shared with other ECUs from the same domain. The challenge in using this approach is that achieving parallelization of ECUs from the same domain cannot be done using typical software parallelization scenarios as used for personal computer software, because of the implied high functional safety demands.

2) *Increasing in-vehicle network communication speed:* As vehicle manufacturers are making an effort to reduce the number of ECUs in vehicles, another challenge arises. In a centralized architecture, the central processing unit can be further away from the data sources it needs to operate correctly. Autonomous cars need to collect and process the surroundings data as fast as possible for their decision making system to react on time. The communication bus system used for gathering data from those sources will have to provide much higher speeds than available in CAN and FlexRay buses of today. Ethernet is emerging as a communication system to potentially answer those needs with its Gigabit speeds [20], but it is still not widely accepted in today's EVs (as seen on Table 3).

3) *Ensuring automotive software reliability:* In CAEVs, software is responsible not only for controlling the vehicle, but for providing new functionalities and driving innovation. As such, the demand for automotive software quality and reliability is very high to guarantee that the vehicles are safe for drivers, passengers and other road users. To ensure reliability, several functional safety standards such as ISO 26262 [67] are in place.

   ISO 26262 is a standard that regulates functional safety of road vehicles and it is the main functional safety standard for the development of software for high-integrity in-vehicle applications [68]. It recommends the use of a Hazard Analysis and Risk Assessment (HARA) method which helps to identify hazardous events in the system and to specify safety goals that mitigate the hazards [69]. Another key requirement of ISO 26262 is the use of Automotive Safety Integrity Level (ASIL) which is a risk classification scheme for an item (e.g. component, software unit) in an automotive system. The ASILs are allocated through a process which covers hazard identification and classification, and risk assessment [70]. Each ASIL depicts the degree of rigor required (e.g. testing techniques, types of documentation required) to reduce the risk of an item [69]. ISO 26262 also provides a specification of safety requirements which need to be allocated to each ASIL to indicate how to prevent or mitigate the hazards.

   ISO 26262 was defined in 2011 and it applies to all systems developed after its publication, meaning that it plays an important role in the development of CAEV automotive software [71]. As CAEVs rely heavily on software for the control and coordination of various components and subsystems, automotive software developers need to understand and implement the standard's requirements, constraints and verification and validation processes to create software which can be used in vehicles.

4) *Improving software security and ensuring privacy protection:* With the implementation of new communication systems arrive new security threats. A connected vehicle with a centralized architecture, can be attacked through the mobile network or Bluetooth [72]. If the vehicle is not prepared to prevent such an attack, the attacker could be able to control the vehicle, potentially putting the lives of the passengers at risk. An attacker could also use the vehicle to retrieve personal information such as the driver's phone directory, call history or GPS coordinates [73]. To prevent the attacker from accessing various nodes of the in-vehicle network, security measures like authentication are required, but a lack of those measures is not uncommon [74]. In electric vehicles, battery management system hacking and intrusion via the charging plug [75] are a reality and security-aware design has to become a major design objective for automotive architectures to prevent these kinds of attacks from happening.

5) *Improving adaptability and usability:* With new security threats potentially being discovered after the deployment of a vehicle on the market, the subsequent addition of new software (OTA updates and patches), and hardware (additional sensors) features will become necessary for fully autonomous vehicles, as their safety will depend on it. Future vehicles will also have to be adaptable to the needs of their passengers. The infotainment systems have to be customizable and provide smartphone integration support for mobile operation systems that come after the vehicle is put on market, not allowing the vehicle OS to become outdated too soon. The integration of mobile phone services such as voicemail, messaging and email into the vehicle's interface makes its usability especially important. Infotainment system needs to allow the driver to focus on driving, while providing seamless access to information and comfort and convenience features [19].

C. IMPACT OF AUTOMOTIVE SOFTWARE

Since its introduction in the 1970s, automotive software's importance in vehicle development has been growing exponentially. Today's CAEVs provide a wide range of infotainment, telematics and driving features, all made possible by automotive software. The next stage of connectedness

TABLE 4. Comparison of the impact of automotive software on different stakeholders.

Stakeholder	Infrastructure owners	Vehicle manufacturers	Vehicle owners	Vehicle users
Role	infrastructure management: deploying EV charging stations, modifying road and telecommunications infrastructure according to the needs of the automotive industry	manufacturing vehicles, developing automotive software	purchasing vehicles, maintaining vehicles	driving, commuting, charging vehicles
Interests	improving the infrastructure, generating profit	selling vehicles and technology, generating profit	purchase and maintenance costs	cost of charging, vehicle efficiency and comfort
Impact of Automotive Software	generating new revenue streams with additional services	generating added value to vehicles, increasing profit	cheapening maintenance by reducing the number of mechanical components in vehicles	reducing charging costs,improving vehicle efficiency and comfort

will bring Software Defined Cars (SDCs) with abilities of over-the-air (OTA) updates capable of changing powertrain capabilities, vehicle dynamics and onboard services, enabling the vehicles to continually adapt to the needs of their passengers [76]. Such changes to the automotive industry, where software is taking over the key role in bringing virtually all new features to life, will impact all stakeholders involved in mobility.

Infrastructure owners are greatly influenced by the developments in the automotive industry. With the number of EVs on the road continually growing, the energy infrastructure owners have to deploy more and more EV charging stations and various research on the methodology of the deployment [77], [78], charging station reservation [79] and willingness-to-pay for EV charging [80] is already being made. Power grid will also have to be adapted to endure the additional charging demand generated by the rising number of EVs [81]. Municipalities and governments responsible for the design, construction, management, safety and maintenance of the road infrastructure were adapting to the mobility needs of the population since the inception of the automobile. When the fully autonomous vehicles become a reality, the road infrastructure will have to be entirely revised. Autonomous vehicle safety demands are stringent, so it will be of the utmost importance to prepare and equip the roads with the technology needed to comply with those demands. If it becomes widely accepted and standardized, the V2I technology will have to be deployed along the existing infrastructure by the municipalities in collaboration with telecommunication companies and vehicle manufacturers. All of this will considerably reduce traffic congestions, making it easier to maintain the roads and infrastructure, saving the municipal money in the process. Telecommunication companies will also have to prepare their infrastructure for the surge in data traffic which will gradually happen when the majority of vehicles on the road become connected to the Internet. Quick and efficient implementation of those changes could bring financial gain to all the aforementioned infrastructure owners.

Vehicle manufacturers are the ones developing new features and deciding the course of automotive software development and the automotive industry as a whole. Selling vehicles is one of the main goals of vehicle manufacturers, and software is one of today's main selling points. This is why it is important for the automotive companies to identify rising trends in the ICT industry as early as possible to be on the forefront of automotive software innovation. Research and development teams play a key part in this process.

Vehicle owners are the ones who are paying for the vehicle to acquire ownership. Personality, lifestyle, travel attitude and mobility factors all affect the individuals' vehicle type choices [82], and software can tailor the vehicle to the needs of the buyer. Smart phones are deeply integrated into today's lifestyle, and an average person today is more tech savvy than ever. Just as the people with a pro-environmental attitude are more likely to buy an electric vehicle, average people are more likely to buy vehicles providing the option of integration of their favourite smart phone brand on the vehicle's infotainment system. Vehicle owners are also responsible for vehicle's maintenance after the purchase. All but essential mechanical components are slowly being replaced by automotive software, making it easier and cheaper for owners to maintain their vehicle.

Recent trends in car sharing and subscription services make it necessary to differentiate vehicle owners and vehicle users. *Vehicle users* are people using the vehicle for transport. This can be a family using a car, passengers in a taxi or truck-drivers driving company provided trucks to transport goods. For a vehicle user, vehicle efficiency and comfort are the key concerns. Energy management systems [83] and intuitive infotainment services are entirely made possible by automotive software, and a major factor in improving those vehicle traits.

A summary of roles and interests of different stakeholders, along with the impact automotive software has on them, can be seen in Table 4.

V. CONCLUSION

This paper provides a review of the automotive software found in today's electric vehicles. In the historical automotive evolution a trend of moving to more sustainable mobility solutions can be seen; ICE powered vehicles are being replaced by their electric counterparts, software has been replacing the mechanical vehicle parts for some time now, connected vehicles are solving traffic congestions in big cities, and autonomous vehicles are on the verge of replacing human drivers. After recognizing those trends, four keywords were identified - electric, softwarized, connected and autonomous - by which a classification of CAEV's software and ICT architecture was made. The identified ICT architecture was then mapped on today's best selling EVs to show the current state of the mentioned automotive technologies. Finally, the trends and challenges in automotive software development were described along with the impact the automotive software has on its stakeholders.

Based on this review, several conclusions can be made. Firstly, electromobility and the introduction of connected, and especially autonomous vehicles, will make automotive software much more important. Not only does software make future automotive advancements possible, it provides added value; the replacement of mechanical parts with software reduces manufacturing costs and the addition of different digital services to the vehicle raises its value in the eyes of the customer. Secondly, the actual complexity of vehicle ICT architecture is much higher than the needed complexity because of its evolutionary development. Electric vehicle manufacturers are revising the existing ICT architecture by taking a holistic approach in product development. The ultimate goal is to reduce the number of ECUs to as low as possible which will lead to faster developing times and reduced costs. Thirdly, the CAEV ICT architecture identified in this paper could change in the coming years as the demands on future connected and autonomous cars become clearer. For example, V2V and V2I technology and their utilization depend entirely on the standardization of communications between different vehicles and the architecture, as well as the willingness of municipalities to employ the technology in their existing road infrastructure. This is why today's CAEVs do not rely on V2V and V2I to achieve their functions. Lastly, as vehicle manufacturers are continuing in their efforts to create a sustainable CAEV, automotive software will serve as a driver for the advancements and the ultimate tool to bring the cost of those vehicles down to acceptable levels for mass production.

This review aims to provide a basis for further research by summarizing the current state, trends and challenges in CAEV automotive software development. The field of CAEV automotive software development requires an interdisciplinary approach as any kind of research or innovation encompasses several different domains. Electric vehicles require knowledge of electrical engineering, connected vehicles require knowledge of telecommunications, while autonomous vehicles require knowledge of computer science and artificial intelligence. Interdisciplinarity is especially important when solving the software and ICT architecture challenges in today's CAEVs. All optimizations of the vehicle ICT architecture, such as reducing its complexity or increasing its communication speed, demand a certain degree of familiarity with vehicle wiring and electronics, as well as communication technologies and software. Creating reliable and secure software cannot be done without the knowledge of underlying hardware and communication protocols. All of this, along with the increasing relevance of automotive software in today's vehicles, the gradual acceptance of EVs and the emergence of connected and autonomous cars, indicates that the number of research projects in the field of CAEVs and CAEV software is sure too grow in the coming years.

REFERENCES

[1] R. Kitchin and M. Dodge, *Code/Space: Software and Everyday Life*. Cambridge, MA, USA: MIT Press, 2011.

[2] E. Juliussen and R. Robinson, "Is Europe in the driver's seat? The competitiveness of the European automotive embedded systems industry," Inst. Prospective Technol. Stud., Eur. Commission, London, U.K., Tech. Rep. JRC 61541, 2010.

[3] M. Broy, "Challenges in automotive software engineering," in *Proc. 28th Int. Conf. Softw. Eng.*, 2006, pp. 33–42.

[4] M. Broy, I. H. Kruger, A. Pretschner, and C. Salzmann, "Engineering automotive software," *Proc. IEEE*, vol. 95, no. 2, pp. 356–373, Feb. 2007.

[5] A. Iwai and M. Aoyama, "Automotive cloud service systems based on service-oriented architecture and its evaluation," in *Proc. IEEE Int. Conf. Cloud Comput. (CLOUD)*, Jul. 2011, pp. 638–645.

[6] M. Di Natale and A. L. Sangiovanni-Vincentelli, "Moving from federated to integrated architectures in automotive: The role of standards, methods and tools," *Proc. IEEE*, vol. 98, no. 4, pp. 603–620, Apr. 2010.

[7] S. Fürst and M. Bechter, "AUTOSAR for connected and autonomous vehicles: The AUTOSAR adaptive platform," in *Proc. 46th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. Workshop*, Jun. 2016, pp. 215–217.

[8] M. Traub, A. Maier, and K. L. Barbehön, "Future automotive architecture and the impact of IT trends," *IEEE Softw.*, vol. 34, no. 3, pp. 27–32, May 2017.

[9] K. Naceur and J. Gagné, "Global EV outlook 2017," Int. Energy Agency, Paris, France, Tech. Rep., 2016.

[10] S. Greengard, "Automotive systems get smarter," *Commun. ACM*, vol. 58, no. 10, pp. 18–20, 2015.

[11] J. Greenough. (2015). *The Connected Car is Creating a Massive New Business Opportunity for Auto, Tech, and Telecom Companies*. [Online]. Available: https://www.businessinsider.com/connected-car-forecasts-top-manufacturers-2015-2

[12] R. Viereckl, D. Ahlemann, A. Koster, and S. Jursch, "Racing ahead with autonomous cars and digital innovation," *Auto Tech Rev.*, vol. 4, no. 12, pp. 18–23, 2015.

[13] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 289–299, Aug. 2014.

[14] R. M. Gandia, F. Antonialli, B. H. Cavazza, A. M. Neto, D. A. D. Lima, J. Y. Sugano, I. Nicolai, and A. L. Zambalde, "Autonomous vehicles: Scientometric and bibliometric review," *Transp. Rev.*, vol. 39, no. 1, pp. 9–28, 2017.

[15] M. R. Endsley, "Autonomous driving systems: A preliminary naturalistic study of the Tesla model S," *J. Cogn. Eng. Decis. Making*, vol. 11, no. 3, pp. 225–238, 2017.

[16] A. A. Alkheir, M. Aloqaily, and H. T. Mouftah, "Connected and autonomous electric vehicles (CAEVs): A service management perspective," *IT Prof.*, vol. 20, no. 6, pp. 54–61, 2018.

[17] R. Rana, M. Staron, N. Mellegård, C. Berger, J. Hansson, M. Nilsson, and F. Törner, "Evaluation of standard reliability growth models in the context of automotive software systems," in *Proc. Int. Conf. Product Focused Softw. Process Improvement*. Berlin, Germany: Springer, 2013, pp. 324–329.

[18] S. Duri, M. Gruteser, X. Liu, P. Moskowitz, R. Perez, M. Singh, and J.-M. Tang, "Framework for security and privacy in automotive telematics," in *Proc. 2nd Int. Workshop Mobile Commerce*, 2002, pp. 25–32.

[19] K. V. Prasad, T. J. Giuli, and D. Watson, "The case for modeling security, privacy, usability and reliability (SPUR) in automotive software," in *Proc. Automot. Softw. Workshop*. Springer, 2006, pp. 1–14.

[20] M. Lukasiewycz, S. Steinhorst, S. Andalam, F. Sagstetter, P. Waszecki, W. Chang, M. Kauer, P. Mundhenk, S. Shanker, S. A. Fahmy, and S. Chakraborty, "System architecture and software design for electric vehicles," in *Proc. 50th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, May/Jun. 2013, pp. 1–6.

[21] M. Brandl, H. Gall, M. Wenger, V. Lorentz, M. Giegerich, F. Baronti, G. Fantechi, L. Fanucci, R. Roncella, R. Saletti, S. Saponara, A. Thaler, M. Cifrain, and W. Prochazka, "Batteries and battery management systems for electric vehicles," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2012, pp. 971–976.

[22] M. Yoshida, O. Tada, and J. Hashime, "Programming of the engine control unit by the C language," Daihatsu Motor Co., Ltd., Ikeda, Japan, SAE Tech. Paper 960047, 1996.

[23] M. Wolf, A. Weimerskirch, and C. Paar, "Security in automotive bus systems," in *Proc. Workshop Embedded Secur. Cars*, 2004, pp. 1–3.

[24] H. Rahimi-Eichi, U. Ojha, F. Baronti, and M.-Y. Chow, "Battery management system: An overview of its application in the smart grid and electric vehicles," *IEEE Ind. Electron. Mag.*, vol. 7, no. 2, pp. 4–16, Jun. 2013.

[25] M. Schmid, "Automotive bus systems," *Automot. Appl.*, pp. 29–32, Dec. 2004.

[26] J. Schäuffele, T. Zurawka, and R. Carey, *Automotive Software Engineering: Principles, Processes, Methods, and Tools*. Wiesbaden, Germany: Springer, 2005.

[27] A. Knirsch, A. Theis, J. Wietzke, and R. Moore, "Compositing user interfaces in partitioned in-vehicle infotainment," in *Proc. Mensch Comput. Workshopband*, 2013, pp. 63–70.

[28] R. Coppola and M. Morisio, "Connected car: Technologies, issues, future trends," *ACM Comput. Surv.*, vol. 49, no. 3, p. 46, 2016.

[29] S. Chakraborty, M. Lukasiewycz, C. Buckl, S. Fahmy, N. Chang, S. Park, Y. Kim, P. Leteinturier, and H. Adlkofer, "Embedded systems and software challenges in electric vehicles," in *Proc. Conf. Design, Autom. Test Eur.*, 2012, pp. 424–429.

[30] A. Sreekanth, K. Srikanth, C. Aditya, T. Satish, and R. Ramchandran, "Deploying common software system for hybrid electric vehicles in AUTOSAR way," in *Proc. IEEE Transp. Electrific. Conf. (ITEC-India)*, Dec. 2017, pp. 1–6.

[31] K. Krügel, L. Stockmann, D. Holler, and K. Lamberg, "Simulation-based development and testing environment for electric vehicles," dSPACE GmbH, Paderborn, Germany, Tech. Rep., 2012, pp. 242–255.

[32] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular ad hoc network," *J. Netw. Comput. Appl.*, vol. 37, pp. 380–392, Jan. 2014.

[33] D. Jiang and L. Delgrossi, "IEEE 802.11p: Towards an international standard for wireless access in vehicular environments," in *Proc. IEEE Veh. Technol. Conf. (VTC Spring)*, May 2008, pp. 2036–2040.

[34] E. Uhlemann, "Initial steps toward a cellular vehicle-to-everything standard [connected vehicles]," *IEEE Veh. Technol. Mag.*, vol. 12, no. 1, pp. 14–19, May 2017.

[35] H. Dakroub, A. Shaout, and A. Awajan, "Connected car architecture and virtualization," *SAE Int. J. Passenger Cars-Electron. Elect. Syst.*, vol. 9, pp. 153–159, Apr. 2016.

[36] K. Jo, J. Kim, D. Kim, C. Jang, and M. Sunwoo, "Development of autonomous car—Part I: Distributed system architecture and development process," *IEEE Trans. Ind. Electron.*, vol. 61, no. 12, pp. 7131–7140, Dec. 2014.

[37] K. Jo, J. Kim, D. Kim, C. Jang, and M. Sunwoo, "Development of autonomous car—Part II: A case study on the implementation of an autonomous driving system based on distributed architecture," *IEEE Trans. Ind. Electron.*, vol. 62, no. 8, pp. 5119–5132, Mar. 2015.

[38] P. Lin, "Why ethics matters for autonomous cars," in *Autonomes Fahren*. Berlin, Germany: Springer, 2015, pp. 69–85.

[39] C. Wilwert, N. Navet, Y.-Q. Song, and F. Simonot-Lion, "Design of automotive X-by-wire systems," Inria, Rocquencourt, France, Tech. Rep. inria-00000562, 2005.

[40] Inside EVs. (2018). *Monthly Plug-in Sales Scorecard*. [Online]. Available: https://insideevs.com/monthly-plug-in-sales-scorecard/

[41] C. Bruce. (2019). *The 10 Best-Selling Vehicles in the United States in 2018 Were Mostly Trucks and SUVs*. [Online]. Available: https://www.motor1.com/features/280320/best-selling-vehicles-us-list/3139164/

[42] Fuel Economy. (2017). *2018 Nissan Leaf*. [Online]. Available: https://www.fueleconomy.gov/feg/Find.do?action=sbs&id=39860

[43] P.-Y. Moulière, A. Chatelain, M. Erriquez, T. Morel, A. Venus, P. Schäfer, and D. Schwedhelm. (2018). *Top Trends in Electric Vehicle Design*. [Online]. Available: https://eu-smartcities.eu/sites/default/files/2018-04/what-a-teardown-of-the-latest-electric-vehicles-reveals-about-the-future-of-mass-market-evs.pdf

[44] J. Allande, P. Tyler, and E. Woodruff, "Nissan leaf on-board diagnostic Bluetooth utility," Cal Polytechn. State Univ., Luis Obispo, CA, USA, Tech. Rep., 2013.

[45] D. Quinn, J. Mitchell, and P. Clark, "Development of a next-generation audible pedestrian alert system for evs having minimal impact on environmental noise levels: Project eVADER," in *Proc. Inter-Noise*, 2014, pp. 1–9.

[46] Nissan. (2018). *NissanConnect EV*. [Online]. Available: https://www.nissan.co.uk/ownership/nissan-infotainment-system/nissanconnect-ev.html

[47] Nissan. (2018). *Nissan Intelligent Mobility*. [Online]. Available: https://www.nissan.co.uk/vehicles/new-vehicles/leaf/intelligent-mobility.html

[48] Nissan. (2017). *Back it Up, Back it in: Nissan's Camera Monitoring Technology*. [Online]. Available: https://www.nissanusa.com/experience-nissan/news-and-events/around-view-monitor.html

[49] C. Paukert. (2015). *What It's Like Riding in a Million-Dollar Autonomous Nissan Leaf*. [Online]. Available: https://www.nissanusa.com/experience-nissan/news-and-events/around-view-monitor.html

[50] Fuel Economy. (2016). *2018 Chevrolet Bolt EV*. [Online]. Available: https://www.fueleconomy.gov/feg/Find.do?action=sbs&id=39786

[51] GM Authority. (2018). *2018 Chevrolet Bolt EV Specs*. [Online]. Available: http://gmauthority.com/blog/gm/chevrolet/chevrolet-bolt-ev/2018-chevrolet-bolt-ev/2018-chevrolet-bolt-ev-specs/

[52] Chevrolet. (2018). *The All-Electric 2018 Bolt EV*. [Online]. Available: https://www.chevrolet.ca/bolt-ev-electric-vehicle.html

[53] OnStar. (2018). *Vehicle Mobile App*. [Online]. Available: https://www.onstar.com/ca/en/mobile_app/

[54] A. J. Hawkins. (2019). *Cruise Postpones Plan to Launch Driverless Taxi Service in 2019*. [Online]. Available: https://www.theverge.com/2019/7/24/20707242/cruise-gm-self-driving-taxi-launch-delay-2019

[55] Fuel Economy. (2017). *2018 Tesla Model S P100D*. [Online]. Available: https://www.fueleconomy.gov/feg/Find.do?action=sbs&id=39840

[56] K. Mahaffey. (2015). *Hacking a Tesla Model S: What We Found and What We Learned*. [Online]. Available: https://blog.lookout.com/hacking-a-tesla

[57] L. Goode. (2017). *ScreenDrive: Tesla Model S is the Epitome of a Tablet on Wheels*. [Online]. Available: https://www.theverge.com/2017/6/19/15827652/2016-tesla-model-s-17-inch-tablet-electric-vehicle-screendrive-review

[58] Tesla. (2018). *Tesla App Support*. [Online]. Available: https://www.tesla.com/en_EU/support/tesla-app

[59] Tesla. (2018). *Tesla Autopilot*. [Online]. Available: https://www.tesla.com/en_EU/autopilot

[60] A. J. Hawkins. (2019). *Here are Elon Musk's Wildest Predictions About Tesla's Self-Driving Cars*. [Online]. Available: https://www.theverge.com/2019/4/22/18510828/tesla-elon-musk-autonomy-day-investor-comments-self-driving-cars-predictions

[61] M. Staron, *Automotive Software Architectures*. Cham, Switzerland: Springer, 2017.

[62] C. Buckl, A. Camek, G. Kainz, C. Simon, L. Mercep, H. Stähle, and A. Knoll, "The software car: Building ICT architectures for future electric vehicles," in *Proc. IEEE Int. Electr. Vehicle Conf. (IEVC)*, Mar. 2012, pp. 1–8.

[63] R. Bertini, H. Wang, T. Knudson, and K. Carstens, "Preparing a roadmap for connected vehicle/cooperative systems deployment scenarios: Case study of the State of Oregon, USA," *Transp. Res. Procedia*, vol. 15, pp. 447–458, Jan. 2016.

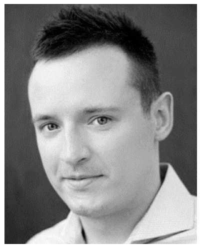
[64] J. M. Lutin, A. L. Kornhauser, and E. Lerner-Lam, "The revolutionary development of self-driving vehicles and implications for the transportation engineering profession," *Inst. Transp. Eng.*, vol. 83, no. 7, pp. 28–32, 2013.

[65] H. Staehle, L. Mercep, A. Knoll, and G. Spiegelberg, "Towards the deployment of a centralized ict architecture in the automotive domain," in *Proc. 2nd Medit. Conf. Embedded Comput. (MECO)*, Jun. 2013, pp. 66–69.
[66] D. Reinhardt and M. Kucera, "Domain controlled architecture," in *Proc. 3rd Int. Conf. Pervasive Embedded Comput. Commun. Syst.*, 2013, pp. 1–6.
[67] *Road Vehicles Functional Safety*, Standard ISO/FDIS International 26262, 2011.
[68] M. Conrad, "Verification and validation according to ISO 26262: A workflow to facilitate the development of high-integrity software," in *Proc. Embedded Real Time Softw. Syst. (ERTS)*, 2012, pp. 1–8.
[69] R. Salay, R. Queiroz, and K. Czarnecki, "An analysis of ISO 26262: Using machine learning safely in automotive software," 2017, *arXiv:1709.02435*. [Online]. Available: https://arxiv.org/abs/1709.02435
[70] D. D. Ward, "System safety in hybrid and electric vehicles," in *Proc. Austral. Syst. Saf. Conf.*, vol. 133, 2011, pp. 79–84.
[71] S. Christiaens, J. Ogrzewalla, and S. Pischinger, "Functional safety for hybrid and electric vehicles," SAE Tech. Paper 2012-01-0032, 2012.
[72] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces," in *Proc. USENIX Secur. Symp.*, San Francisco, CA, USA, 2011, pp. 77–92.
[73] I. Studnia, V. Nicomette, E. Alata, Y. Deswarte, M. Kaaniche, and Y. Laarouchi, "Survey on security threats and protection mechanisms in embedded automotive networks," in *Proc. 43rd Annu. IEEE/IFIP Conf. Dependable Syst. Netw. Workshop (DSN-W)*, Jun. 2013, pp. 1–12.
[74] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, and H. Shacham, "Experimental security analysis of a modern automobile," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2010, pp. 447–462.
[75] F. Sagstetter, M. Lukasiewycz, S. Steinhorst, M. Wolf, A. Bouard, W. R. Harris, S. Jha, T. Peyrin, A. Poschmann, and S. Chakraborty, "Security challenges in automotive hardware/software architecture design," in *Proc. Conf. Design, Autom. Test Eur.*, 2013, pp. 458–463.
[76] Ericsson. (2018). *Software Defined Cars*. [Online]. Available: https://www.ericsson.com/en/internet-of-things/trending/software-defined-cars
[77] D. Pevec, J. Babic, M. A. Kayser, A. Carvalho, Y. Ghiassi-Farrokhfal, and V. Podobnik, "A data-driven statistical approach for extending electric vehicle charging infrastructure," *Int. J. Energy Res.*, vol. 42, no. 9, pp. 3102–3120, 2018.
[78] J. Babic, A. Carvalho, W. Ketter, and V. Podobnik, "Evaluating policies for parking lots handling electric vehicles," *IEEE Access*, vol. 6, pp. 944–961, 2018.
[79] B. Vaidya and H. T. Mouftah, "Automated reservation mechanism for charging connected and autonomous EVs in smart cities," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–5.
[80] L. Dorcec, D. Pevec, H. Vdovic, J. Babic, and V. Podobnik, "How do people value electric vehicle charging service? A gamified survey approach," *J. Cleaner Prod.*, vol. 210, pp. 887–897, Feb. 2019.
[81] S. Habib, M. M. Khan, F. Abbas, L. Sang, M. U. Shahid, and H. Tang, "A comprehensive study of implemented international standards, technical challenges, impacts and prospects for electric vehicles," *IEEE Access*, vol. 6, pp. 13866–13890, 2018.
[82] S. Choo and P. L. Mokhtarian, "What type of vehicle do people drive? The role of attitude and lifestyle in influencing vehicle type choice," *Transp. Res. A, Policy Pract.*, vol. 38, no. 3, pp. 201–222, 2004.
[83] S. F. Tie and C. W. Tan, "A review of energy sources and energy management system in electric vehicles," *Renew. Sustain. Energy Rev.*, vol. 20, pp. 82–102, Apr. 2013.

HRVOJE VDOVIC (GS'18) received the M.Sc. degree in information and communication technology from the Faculty of Electrical Engineering and Computing (FER), University of Zagreb, Zagreb, Croatia, in 2017, where he is currently pursuing the Ph.D. degree with the University of Zagreb.

He is currently a Research Associate with the Social Networking and Computing Laboratory, Telecommunication Department, FER. His teaching activities are in the field of software development and network science. His current research is focused on software development and bus message simulation for electric and connected vehicles. He is actively involved as a researcher and a developer on several industrial and scientific projects.

Mr. Vdovic is a member of IEEE International Association.



JURICA BABIC (GS'12–M'18) received the M.Sc. degree in computing and the Ph.D. degree in computer science from the Faculty of Electrical Engineering and Computing (FER), University of Zagreb, Zagreb, Croatia, in 2012 and 2018, respectively.

Since 2012, he has been with the Department of Telecommunications, FER, where he has been an Assistant Professor, since 2019. He is the Energy Informatics Team Leader of the Social Networking and Computing Laboratory and a Researcher with the Laboratory for Assistive Technologies and Augmentative and Alternative Communication. He is actively involved in research and development activities with several national and international scientific and industrial projects. His teaching and research activities are in transdisciplinary fields of network and data science, electromobility, and augmentative and alternative communication. He has coauthored over 30 articles, including publications in IEEE Access, the *International Journal of Energy Research*, and the *Journal of Cleaner Production* journals.

Prof. Babic is a member of the ACM, INFORMS, and KES International Associations. He was a member of an interdisciplinary team which was a recipient of the Highest National Award for notable achievements in the education activity by the Croatian Parliament, in 2015, and the Annual National PMI Project of the Year Award by the world's leading project management professionals association PMI, in 2016.

VEDRAN PODOBNIK (M'05–SM'17) received the M.Eng. degree in electrical Engineering and the Ph.D. degree in computer science from the University of Zagreb, Faculty of Electrical Engineering and Computing (FER), Zagreb, Croatia, in 2006 and 2010, respectively, and the M.Phil. degree in technology policy from the University of Cambridge, Judge Business School, Cambridge, U.K., in 2013.

He works as an Associate Professor with the Department of Telecommunications, FER, where he is also the Founder and Director of the "Social Networking and Computing Laboratory (socialLAB)." He is a Scientific Advisor for data science to a leading global IT company, HPE. He has led several national and international scientific and industrial projects. He is currently a principal investigator in two research projects dealing with data science aspect of future transportation systems based on e-mobility (financed by the EU) and a Work Package Leader of the only National Center of Research Excellence (CoRE) in the field of technical sciences, the "CoRE for Data Science and Cooperative Systems." His teaching and research activities are in transdisciplinary fields of network and data science, e-mobility, and technology policy. He has coauthored over 80 articles, including publications in the *Information Technology & People*, the *International Journal of Energy Research*, the *Journal of Cleaner Production*, and the *AI Magazine* journals.

Prof. Podobnik is a member of the ACM, INFORMS, AIS, and KES International associations. He has participated in more than 30 conference international programs committees, and he serves as a Technical Reviewer for various international journals. He was the leader of an interdisciplinary team which was awarded the highest national award for notable achievements in the education activity, in 2015, awarded by the Croatian Parliament, as well as the annual National PMI Project of the Year Award, in 2016, awarded by the world's leading project management professionals association PMI. As a Junior Researcher, he received the Croatian Annual National Award for Science in the field of technical sciences, in 2011, awarded by the Croatian Parliament, as well as the Silver Medal "Josip Loncar" Award for outstanding doctoral dissertation and particularly successful scientific research, in 2010, awarded by FER.