A Survey of Reference Architectures for Autonomous Cars

BIJAYITA THAPA, Florida Atlantic University
EDUARDO B. FERNANDEZ, Florida Atlantic University

Autonomous cars are getting attention from every sector. However, as their popularity increases, the number of users increases as well, including potential hackers, and they could become the target of cyber-attacks. Before we study how to enumerate their security threats, it is essential to build a precise reference architecture (RA). An RA describes at an abstract level the functionality of these vehicles. Such RAs can be analyzed to find their threats and will lead us to define countermeasures for those threats and to have a better understanding of the security of autonomous cars. We studied several proposed RAs for autonomous cars, and after analyzing and comparing them, we found that these architectures lack precision and detail, which makes the analysis and identification of possible vulnerabilities and threats inadequate (not specific enough, not complete). From this comparison we have defined an outline for an improved model.

Categories and Subject Descriptors: D.2.11 **[Software Engineering]** Software Architectures–Patterns;

General Terms: Design

Additional Key Words and Phrases: Autonomous car, security patterns, use cases, secure architecture

ACM Reference Format:

Thapa, B. and Fernandez, E. B., A Survey of Reference Architectures for Autonomous Cars. Procs. of the 27th Conf. on Pattern Lang. of Prog. (Plop'20), October 2020, 10 pages.

1. INTRODUCTION

The world is undergoing enormous changes, such as dramatic population growth, urbanization, resource shortages, and global warming. According to (Hui et al. 2017), 66% of the world population will be living in urban areas by 2050. As the population keeps increasing, the number of cars is also increasing, especially in the urban areas, causing accidents, traffic congestion, environmental pollution, etc. One of the solutions for all these problems is the use of autonomous or self-driving cars. However, autonomous cars have a large attack surface and we are considering how to improve their security. Our objective is to build a Security Reference Architecture (SRA) where threats can be assigned to specific units and where security defenses can be added to stop these threats. An SRA is an abstract architecture where possible threats have been identified.

The architecture design for autonomous cars is equivalent to the design of a real-time, mobile, intelligent, control system (Serban et al. 2018). An autonomous car includes techniques to detect and process its surrounding environment, such as GPS, laser, radar, lidar, odometry, computer vision, artificial intelligence (AI), cameras, and others. The presence of such technologies allows it to interpret sensory information, identify its location, navigate paths, avoid obstacles and congestion, and follow relevant signs. Autonomous cars must be able to differentiate between various types of vehicles, pedestrians, and other obstacles on the road, by analyzing the data collected by sensors. Also, AI influences the technology in the vehicles; for example, AI helps the cameras in the vehicle to identify the people in that vehicle, track their eye position and decide whether that person is falling asleep, tired, or distracted. A fully autonomous car is able to perceive, make decisions, and plan.

Also, autonomous cars can be connected with each other (V2V), with the surrounding infrastructure (V2I), and with other various devices (V2X). The increasing level of connectivity and automation of vehicles enhances safety, produces efficient traffic management, allows receiving weather advisory information, provides entertainment, etc.

Author's address: Bijayita Thapa (corresponding author), Dept. of Computer and Electrical Eng. and Computer Science, Florida Atlantic University, 777 Glades Rd., Boca Raton, FL33431, USA; email: bthapa@fau.edu; Eduardo B. Fernandez, Dept. of Computer and Electrical Eng. and Computer Science, Florida Atlantic University, 777 Glades Rd., Boca Raton, FL33431, USA; email: fernande@fau.edu

At the same time, as the connectivity and level of automation increases, there is a higher risk of cyber-attacks that can lead to crashes and other problems. Many attacks used to be only possible with physical access to a vehicle, but now they can be carried out remotely through the wireless networks used in autonomous cars and their connection to the internet (Zhang et al. 2014). There can be a risk for both intra-vehicle security (the system and various components within the car can be compromised) and inter-vehicle security (the car can be compromised when there is V2V, V2I, and V2X connections). Since autonomous cars are driverless, security and safety are very critical. Their security has impact on safety, thus (Glas et al. 2014) started building safety-relevant systems with security requirements. Sometimes, the security consequences lead to safety consequences, such as injuries and loss of human life.

The objective of this survey is to understand and compare the proposed RAs for autonomous cars. Once we have a strong understanding of these RAs, we can develop a new RA incorporating the best features found in previous RAs. Our RA will later be used to analyze possible vulnerabilities and threats, and develop misuse and security patterns as countermeasures. The currently available RAs are not precise and detailed enough for this purpose, and also most of them do not list their use cases. This requires a new and more precise RA. Our RA will be built using UML to define stakeholders, use cases, and functional patterns. Later, we intend to build an SRA by analyzing threats and adding security patterns to the RA.

The remainder of this paper is organized as follows. Section 2 describes some background that includes patterns, RAs, and SRAs. This section also includes general concepts of autonomous cars, as well as their components. Section 3 is a survey of RAs for autonomous cars. Section 4 compares the various RAs in a tabular form and presents some guidelines to build an improved RA. We end with conclusions in section 5.

2. BACKGROUND

Along with its advances in software and technology, the automotive industry is also producing more complex vehicles. According to (Serban et al. 2018), even a basic car has large amounts of software with tight constraints concerning real-time processing, failure rate, maintainability, and safety. An autonomous car will be a very complex system by itself and even more when including its ecosystem. Due to this complexity and level of integration, cars may malfunction because of various security attacks; in other words, they have a large attack surface. A critical issue in the design and construction of any complex software system is its architecture (Garlan 2000), (Fernandez 2013). By using patterns, RAs, and SRAs during their development stage, we can handle this level of complexity.

2.1 Patterns and Reference Architectures

A pattern is a solution to a recurrent software problem in a given context (Buschmann 1996, Fernandez 2013). A pattern can be used as a guideline for design and captures the experience and knowledge of designers to develop models that can be used for new designs (Fernandez 2013). There are various types of patterns, such as architecture patterns, security patterns, misuse patterns, and others. A security pattern describes a solution to a security problem combining knowledge about security with the structure provided by patterns. It also describes a precise generic model for a security mechanism and is a basic artifact for a secure development methodology (Uzunov et al. 2015). A misuse pattern describes how a misuse is performed from the view point of an attacker, what system units it uses and how, and provides ways of stopping the attack by applying possible security patterns (Fernandez 2013).

An RA is an abstract architecture that describes the functionality of a system at a high level of abstraction without containing implementation details (Avgeriou 2003). According to the DoD Reference Architecture Description (CIO 2010), an RA is an authoritative source of information about a specific subject area that guides and constrains the instantiations of multiple architectures and solutions. Moreover, an RA provides a common language for the various stakeholders, consistency of implementation of technology, support for the validation of solutions, and encouragement to adhere to common standards and specifications (CIO 2010). An RA is a useful tool to understand and build a complex system by describing its main components at an abstract level (Fernandez et al. 2016). An RA is composed of several analysis patterns taken from some domain and can be seen as a compound pattern made of several smaller patterns that model specific aspects of the architecture. It can also be seen as a template-like solution that can be instantiated into a concrete software architecture by adding implementation-oriented aspects. Our work

on Semantic Analysis patterns (Fernández & Yuan 2000), was a step towards building RAs out of patterns, an idea also present in (Muller and de Laar 2009) and (Stricker et al. 2010).

An SRA is an abstract architecture that includes patterns to realize security or privacy requirements. SRAs describe a conceptual model of security and provide a way to specify security requirements for a wide range of concrete architectures (Fernandez et al. 2016). Also, they are useful to guide the security design of systems by providing generic solutions that can stop a variety of attacks (Fernandez et al. 2009). The use of patterns and RAs are a powerful way to organize and describe security and other non-functional aspects, such as privacy, safety, performance, reliability, and availability (Romero & Fernández 2017). Also, a good architecture can help to make sure that a system will satisfy key requirements in areas such as performance, reliability, portability, scalability, and interoperability (Garlan 2000). In particular, safety is essential for any type of car, and requires reliability, availability, and security aspects.

2.2 Autonomous Cars

As indicated, when vehicles get more advanced, their complexity and connectivity also increase and become more prone to security attacks. There has been tremendous progress and work done on autonomous cars, but relatively little work has been done for their security and privacy issues.

An autonomous car includes various techniques to detect its surroundings. Some of the components of the autonomous cars are the following:

- *Event Data Recorder (EDR)* – it is like the black boxes found in airplanes; they can record all the major data for crash reconstruction.
- *Sensors and Cameras* – an autonomous car is loaded with various types of sensors and cameras. Sensors and cameras will allow it to have functionalities such as lane-keep assist, electronic payment for toll, automatic parking, etc. Also, sensors are needed to provide an autonomous system with situational awareness about the physical world (Wyglinski et al. 2013). The main sensing devices in the autonomous car are:
    - Radars - Front-end radar is used to detect obstacles at a distance.
    - Lidars - Lidar sensors generate maps that can be used for localization, obstacle avoidance, and navigation.
    - Ultrasound sensors – used for parking assistance. These sensors help to monitor the front and rear of the car and also warn if there is any obstacle.
    - Laser scanners – these are used for obstacle detection when there are limitations with camera-based perception (Jo et al. 2013).
    - Cameras - When cameras are fused with sensors, they can provide static and dynamic obstacle detection, object recognition, and 360-degree surrounding information (Wyglinski et al. 2013).
- *Global Positioning System (GPS)* – this system is mainly used for localization by measuring the position of the car.
- *Electronic Control Units (ECUs)* – Every function performed by autonomous cars is handled by ECUs. ECUs are responsible for many operations, such as power locks, seat adjustments, power steering, brakes, fuel injection, and emissions control (Jo et al. 2013). ECUs are responsible for data communications between various components of the car. They also handle communications among cars, and the car and roadside infrastructure or other devices.
- *Artificial Intelligence (AI) software* – AI influences the technology in the cars because it implies the ability of a program to perceive, plan, learn and make decisions. Autonomous cars will produce enormous amounts of data that needs to be interpreted.

The hardware and software in autonomous cars can be arranged into various layers based on their functionality. The hardware includes the sensor layer and the computation layer. The sensor layer is for perception, which includes components, such as sensors, cameras, and scanners to collect data around its surroundings and within the car itself. In the sensor layer, data is collected all the time with a 360-degree scope. The computation layer includes CPU, GUP, and other related functions. The software includes the interface layer and the application layer. Sensor fusion in the

interface layer takes all the signals from the sensor layer, processes them, and performs planning. The software in the application layer makes decisions based on the planning provided by the interface layer. Also, AI is an important component for processing data, planning, and decision making.

3. REFERENCE ARCHITECTURES OF AUTONOMOUS CARS

It is clear that given the complexity of car systems, a distributed architecture is more suitable than a centralized one. When looking for RAs we found that the terms "reference architecture" appears with variations such as "functional view", functional architecture view" (Behere & Törngren 2016), or system architecture (Jo et al. 2013). According to (Jo et al. 2014) and (Jo et al. 2015), the basic functional components of the autonomous car are divided into:

- Perception – the autonomous car is capable of sensing its surrounding environment. For this function, it can use various types of sensors.
- Localization – this functionality finds the position of the autonomous car and for this function, Global Positioning System (GPS), dead reckoning, and road maps are used.
- Planning – this functionality determines the behavior of the car based on the information perceived by the car and its localization.
- Vehicle Control – this function includes steering, accelerating, and braking. These are regulated by the planning function.
- System management – this system manages the fault management system, logging system, and human-machine interface (HMI).

(Behere & Törngren 2016) presented a functional architecture based on the concepts described in the ISO 26262 automotive functional safety standard (Serban et al. 2018). They divided architecture into two layers, vehicle platform and cognitive driving intelligent. Also, they divided the functional components of each layer of autonomous cars into three main categories:

- Perception – divided into sensing, sensor fusion, localization, semantic understanding, and world model.
- Decision and control – include trajectory generation, energy management, diagnosis and fault management, reactive control, and vehicle platform abstraction.
- Vehicle platform manipulation – include platform stabilization, passive safety, and trajectory execution. Trajectory execution includes propulsion, steering, and braking.

While designing their functional architecture, they have taken in consideration stakeholder concerns. They have categorized these concerns into business concerns and engineering concerns.

(Jo et al., 2013) presented an RA categorizing the system architecture into four layers, where each layer is composed of several components (Fig. 1). The higher-level components layer is composed of localization, perception, planning and vehicle control. The in-vehicle network layer is composed of FlexRay (an automotive network protocol), Controller Area Network (CAN), Ethernet and Gateway to facilitate communication among nodes in the high-level and the low-level component layers. The low-level components layer includes nodes for various functionalities, such as object detection, vision, positioning, vehicle state estimation, steering control, and acceleration control. The sensors and actuators layer includes laser scanners, camera, GPS, inertial measurement unit (IMU), and other various components. Components of the sensors and actuators layer perform the various functionalities in the low-level components layer. For example, laser scanners are used for obstacle detection while cameras are used for object detection such as passengers, crosswalks, traffic lights, and any other obstacles around the car's surroundings. Similarly, IMU is used for the integrity and accuracy of the position.

The authors of (Maple et al. 2019) believe that most of the RAs do not include the complete range of interactions between autonomous cars, devices and peripherals, edge and the cloud. Therefore, they presented an RA which is composed of four sub-architectures, namely sub-architectures for connected autonomous cars and devices and peripherals, sub architecture for edge, and sub architecture for cloud (Fig. 2). However, figures and descriptions on sub-architecture for edge and sub-architecture for cloud are not included in the paper. In general, the existing RAs to connect autonomous cars consider analysis of attacks and risks, viewpoints and features of autonomous cars, devices, edge and cloud. There are various viewpoints that can be presented in an RA, such as Functional, Communication, Implementation, Enterprise, Usage, Information, and Physical. In Fig. 2, they have used the Functional and

Communication viewpoints and combined them into a single hybrid viewpoint. The components included in this RA are wireless communication, physical inputs/outputs, internal (virtual) communication, sensors, data storage, actuation, monitoring and logging, and others as shown in the Fig. 2.

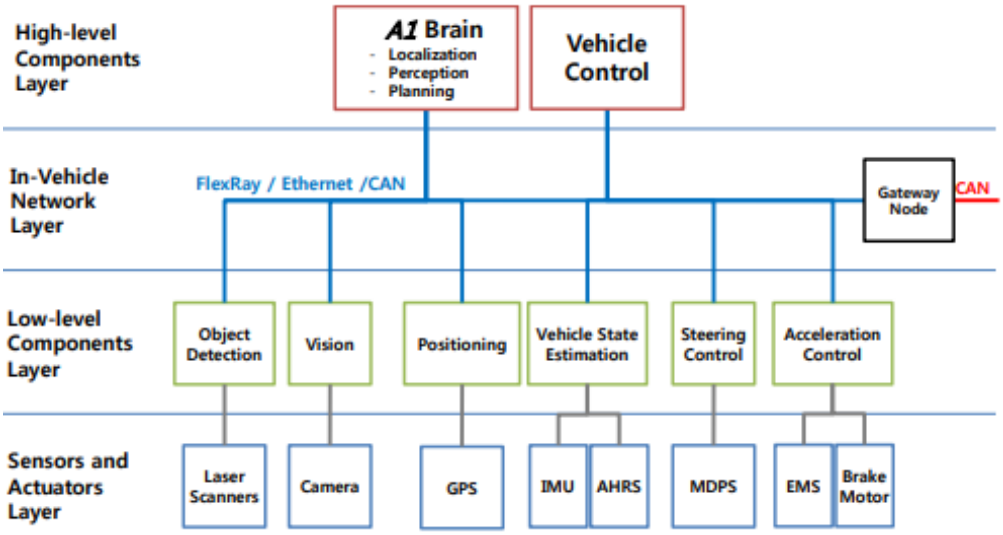

Fig. 1. Distributed System Architecture of Autonomous Car (Jo et al. 2013)

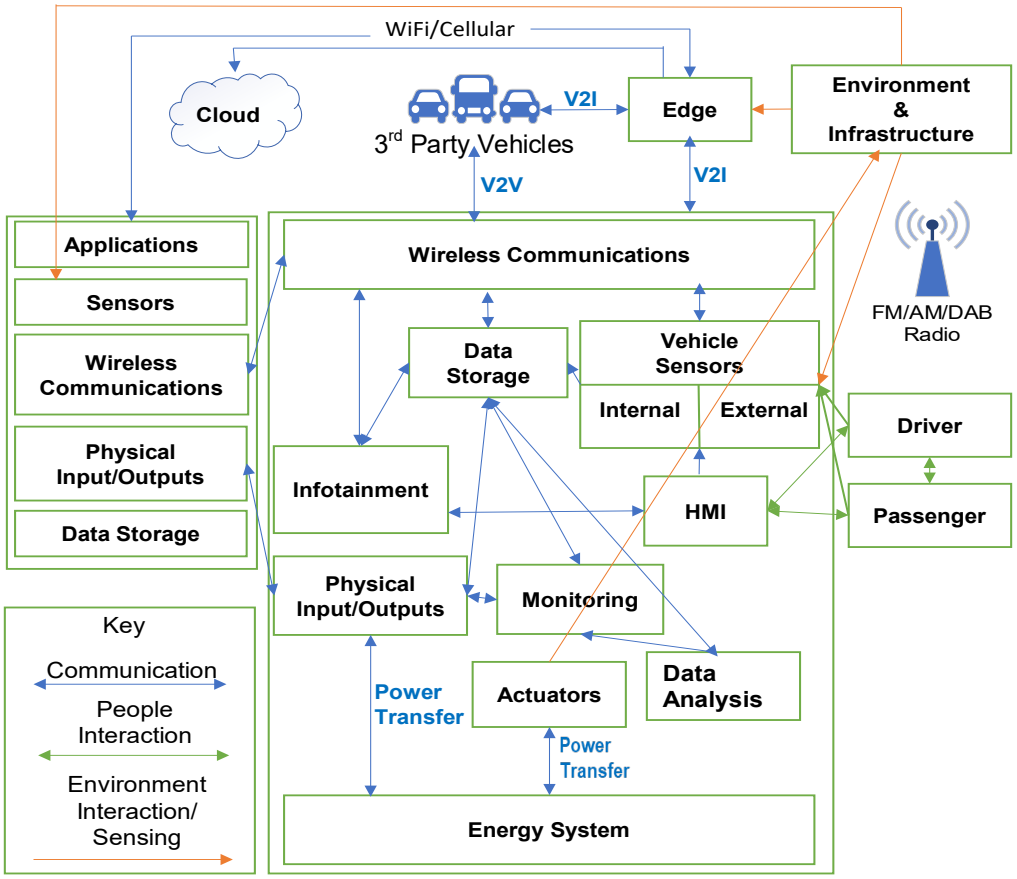

Fig. 2. Connect Autonomous Cars and Devices and Peripherals Reference Architecture (Maple et al. 2019)

The RA presented in (Pelliccione et al. 2020) is composed of three different layers having the three different perception capabilities of a car – internal layer, external layer for vehicle surrounding, and external for system of systems. The internal layer has the capabilities for sensing within the car, whereas the external layer has the capabilities to sense the surrounding of the car. Instance camera, radars, and lidar are used for sensing, and data collected from these sensors are used to produce the needed information. The External Layer for System of Systems (SoS) enables the vehicle to receive information from other cars, street signs, traffic lights, parking, pedestrians, cyclists, etc., through various communication means that are beyond the limits of sensors and devices present in the car. Each layer includes components for various functionalities as shown in Fig. 3.

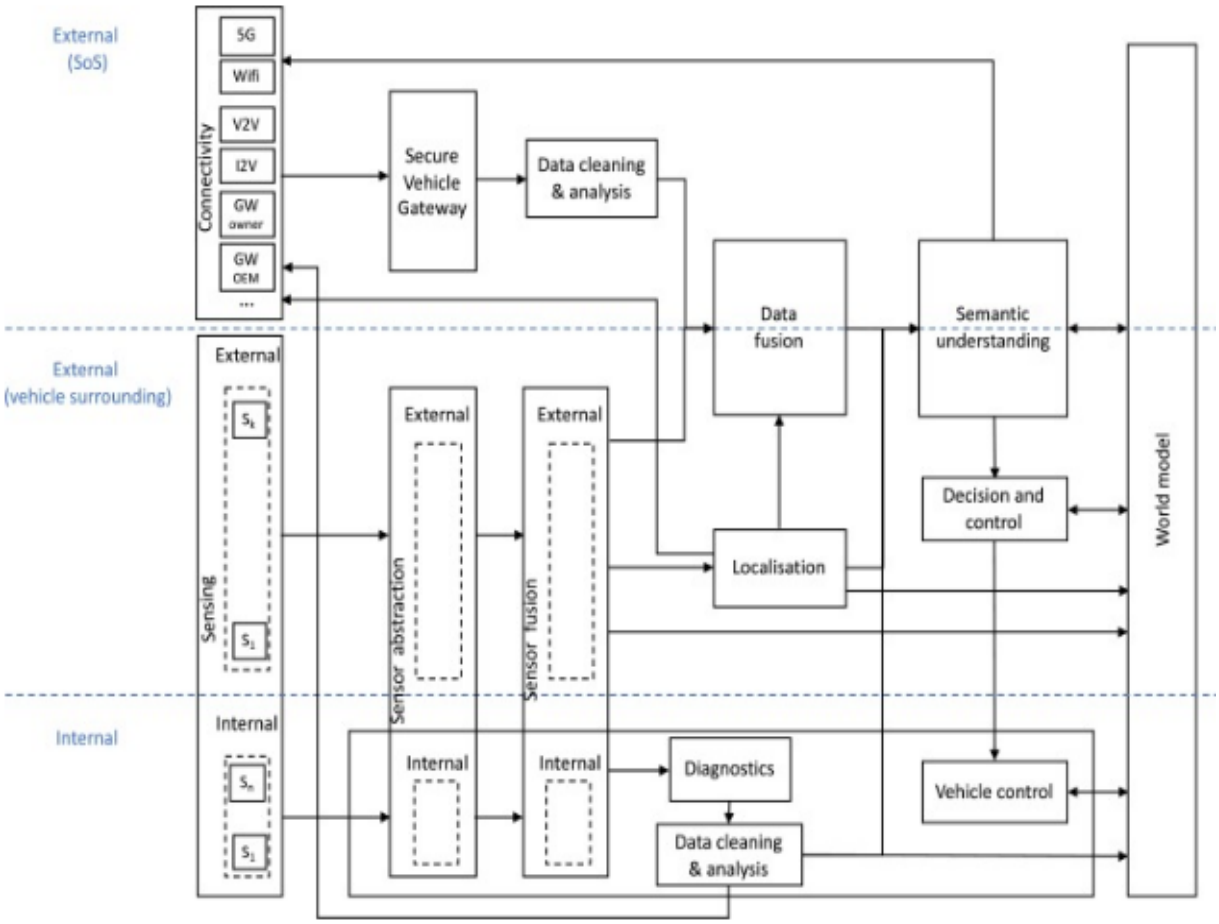

Fig. 3. Functional reference architecture for vehicles as parts of a SoS (Pelliccione et al. 2020)

(Plappert et al. 2017) developed an RA that focuses on describing the relevant data flows and various interfaces regarding privacy and to generalize from the solutions of different manufactures. In Fig. 4, these interfaces are classified into three different classes, such as on-board diagnostics (OBD) port, wireless interfaces, and physical ports. OBD port is used for accessing data from the electronic control units in the car. The wireless interfaces include Bluetooth, Wi-Fi, Cellular, or V2V communication. The physical ports include CD/DVD slots, USB or SD ports, which are mainly used for inserting data into the car or extracting data from the car. Besides, they partitioned a car into various logical domains, such as Powertrain, Driving Dynamics, and others. These domains consist of various components like ECUs, actuators, and sensors. Then, they developed a privacy-aware data access system for automotive applications to protect privacy and control third-party access. The system was developed using ten relevant use cases in six categories. The U.S. Department of Transportation (USDOT) has worked on a project called Connected Vehicle Reference Implementation Architecture (CVRIA) (U. S. Department of Transportation 2019), that

has developed a framework for the integration and standardization of connected vehicles. CVRIA is composed of four viewpoints as listed below:

- Enterprise viewpoint – which describes the relationship between organizations and their roles with a connected vehicle environment
- Functional viewpoint – the abstract functional elements, i.e., processes and their logical interactions such as data flows
- Physical viewpoint – physical objects, such as systems and devices, application objects, and the high-level interfaces between physical objects
- Communication viewpoint – the protocols needed to support communications among physical objects in the connected car ecosystem.

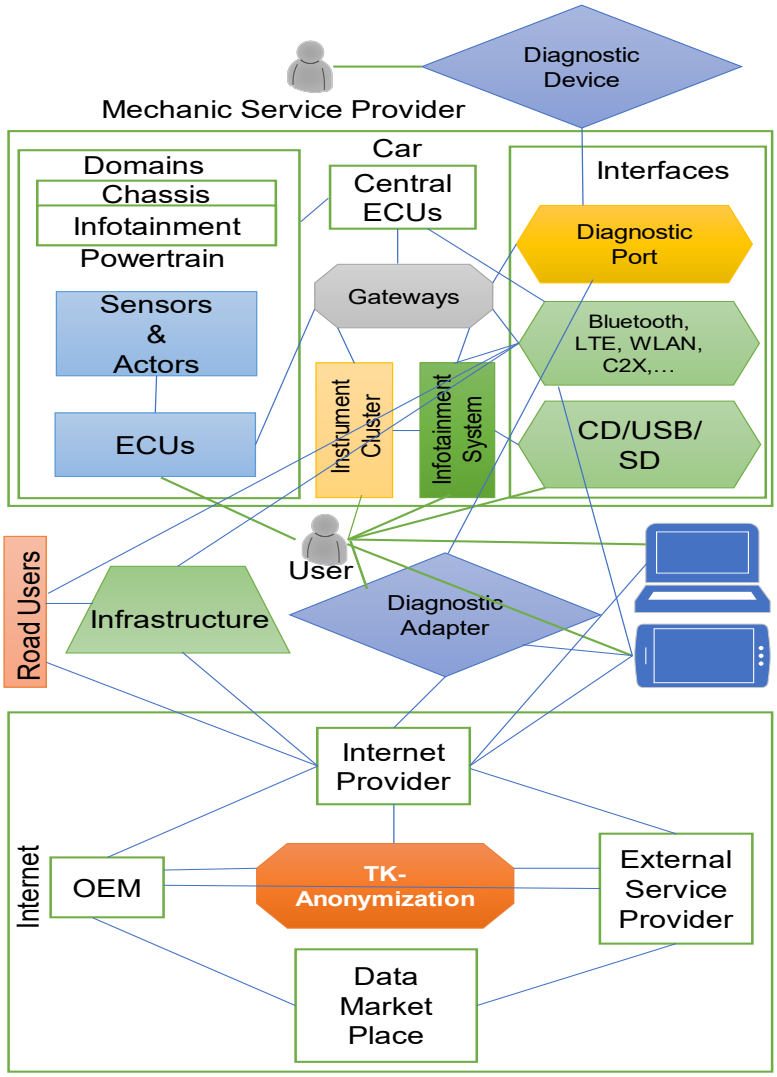

Fig. 4. Connected Car RA (Plappert et al. 2017)

(Dominic et al. 2016) believe that a high level of automation for driving is still in the development phase; therefore, there is no standard architecture for autonomous cars. To solve this problem, they developed a reference architecture that consists of two levels, i.e., higher-level blocks and lower-level blocks as shown in Fig. 5. The higher-level blocks include general applications, such as maps, sensors, vehicle interfaces, sensor fusion and processing, control, and external communications. The lower-level blocks include more specific applications, such as road network map, GPS, actuation, etc.

They divided the sensor fusion and processing application into three modules:

- Localization – this module localizes the car in its environment by using odometry sensors, range sensors, a GPS receiver, etc.
- Object detection – this module is intended to detect, classify, and track objects in the surrounding environment.
- Path planning – this module is for planning trajectories by using the output from localization and object detection.

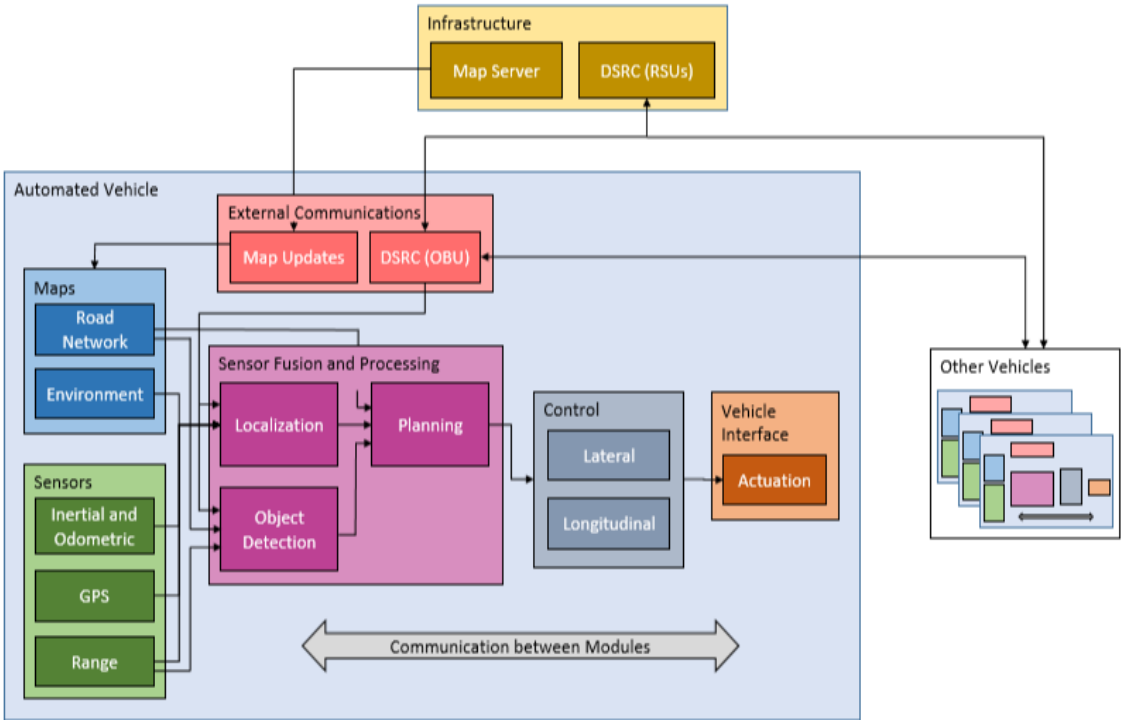

Fig. 5. Interconnected Automated Driving RA (Dominic et al. 2016)

4. COMPARING REFERENCE ARCHITECTURES

After studying the various architectures shown above, we analyzed and presented them in Table 1. All of these RAs are described using block diagrams that are not good enough for our future goal, i.e., to enumerate threats in autonomous cars. Only (Plappert et al. 2017) has considered use cases, but they are described without much detail. We also found that most of the works are mainly focused on specialized aspects, but security requires a holistic global view of the system. None of them indicates explicitly any patterns but it is possible to identify some patterns in these models. For example, all these architectures have sensors and there is already in the literature a Sensor Node pattern (Sahu et al. 2010). Another pattern present in all architectures is the Actuator, now under development by our group.

This pattern will be part of our next stage; the new and old patterns can be combined to define a better RA. We can also identify their use cases and from them develop more details of autonomous cars architectures. The RA can be made even more generic by defining variability patterns that can be configured to build an extensible RA that can describe different types of autonomous vehicles. Secure versions of the patterns of the RA; for example, the Secure Sensor Node (Orellana et al. 2020), could then be combined to form a SRA.

Table 1 Summary of RAs for Autonomous Cars

ID	Reference Architecture	Standard Used	Main Components (layers) in the RA	Viewpoints	Scope	Safety, Security, or Privacy	Analysis Using Use Cases
RA1	(Jo et al. 2014), (Jo et al. 2015)	AUTOSAR		Technical	In-vehicle components only	None	No
RA2	(Behere & Törngren 2016)	ISO 26262 (for safety)	1. Perception 2. Decision & Control 3. Vehicle platform manipulation	Functional Enterprise	In-vehicle components only	None	No
RA3	(Jo et al. 2013)		1. High-level components layer 2. In-vehicle network layer 3. Low-level components layer 4. sensors and actuators layer		In-vehicle components only	None	No
RA4	(Maple et al. 2019)			Functional Communication	In-vehicle components, connected devices, edge and cloud	Security (attack analysis)	No
RA5	(Pelliccione et al. 2020)	ISO/IEC/IEEE 42010:2011	1. Perception 2. External (vehicle surrounding) 3. External (system of systems)	Architecture (system of systems)	In-vehicle components and external components	None	No
RA6	(Plappert et al. 2017)	ISO 9241-210 Legal framework - GDPR	1. Domains 2. Interfaces 3. Internet		In-vehicle components and external components	Privacy	10 use cases used for 6 different categories, described in brief without UML diagrams
RA7	U.S. DoT (U. S. Department of Transportation 2019)			Enterprise Functional Physical Communication	In-vehicle components and external components	Security	No
RA8	(Dominic et al. 2016)		1. Higher-level blocks 2. Lower-level blocks		In-vehicle components and external components	Security (risk assessment)	No

5. CONCLUSIONS

Autonomous cars are still in their development phase. However, when autonomous cars become a product and are available in the market, we will find them everywhere. There will be V2V, V2I, and V2X connections, which will make autonomous cars have a large attack surface. To improve their security, we studied several proposed RAs and compared them based on the standards used, their main components and viewpoints, security and privacy, and their use cases. After looking at many references, we found them not appropriate to define security aspects because of their lack of precision and details. Our next step is to develop an improved version of an RA, including its stakeholders and its use cases, according to the guidelines defined in Section 4 and following the ideas of (Avgeriou 2003). Then we can enumerate their vulnerabilities and threats as well as their defenses leading to a SRA.

REFERENCES

P. Avgeriou, "Describing, Instantiating and Evaluating a Reference Architecture: A Case Study," Enterprise Architecture Journal, June 2003.

S. Behere and M. Törngren, "A functional reference architecture for autonomous driving," Information and Software Technology 73, 2016, 136-150.

F. Buschmann, R. Meunier, H. Rohnert, P. Sommerland, and M. Stal, Pattern-Oriented Software Architecture Volume 1: A System of Patterns, Volume 1. Wiley, 1996.

CIO, "DoD Reference Architecture White Paper," June 2010. Retrieved July 13, 2019 from https://dodcio.defense.gov/Portals/0/Documents/DIEA/Ref_Archi_Description_Final_v1_18Jun10.pdf

D. Dominic, S. Chhawri, R. M. Eustice, D. Ma, and A. Weimerskirch, "Risk Assessment for Cooperative Automated Driving," CPS-SPC '16: Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy, 2016, Vienna, Austria.

E. B. Fernandez, "Security Patterns in Practice: Designing Secure Architectures Using Software Patterns," Hoboken: John Wiley & Sons, 2013.

E. B. Fernandez, R. Monge, and K. Hashizume, "Building a security reference architecture for cloud systems," *Requirements Engineering*., June 2016, Volume 21, Issue 2, 225-249. Doi: 10.1007/s00766-014-0218-7

E. B. Fernandez, N. Yoshioka, and H. Washizaki, "Modeling Misuse Patterns," 2009 International Conference on Availability, Reliability and Security, 566-571, Fukuoka, Japan: IEEE.

E. B. Fernandez and X. Yuan, "Semantic Analysis Patterns", *Procs. 19th Int. Conf. on Conceptual Modeling*, ER2000, Salt Lake City, UT, October 2000. *Lecture Notes in Computer Science*, Volume: 1920, 183-195

D. Garlan, "Software architecture: a roadmap," ICSE '00 Proceedings of the Conference on The Future of Software Engineering, 2000, Limerick, Ireland.

B. Glas, C. Gebauer, J. Hänger, A. Heyl, J. Klarmann, S. Kriso, P. Vembar, and P. Wörz, "Automotive safety and security integration challenges," In: H. Klenk, H. B. Keller, E. Plödereder, and P. Dencker (Hrsg.), Automotive - Safety & Security 2014. Bonn: Gesellschaft für Informatik e.V (S. 13-28).

T. K. Hui, R. S. Sherratt, and D. Díaz-Sánchez, "Major requirements for building Smart Homes in Smart Cities based on Internet of Things technologies," Future Generation Computer Systems, 2017, 76, 358-369.

K. Jo, J. Kim, D. Kim, C. Jang, and M. Sunwoo, "Development of Autonomous Car—Part I: Distributed System Architecture and Development Process," IEEE Transactions on Industrial Electronics , 61(12), 2014, 7131-7140.

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 Transactions on Industrial Electronics, 62(8), 2015, 5119-5132.

K. Jo, M. Lee, D. Kim, J. Kim, C. Jang, E. Kim, and M. Sunwoo, "Overall Reviews of Autonomous Vehicle A1-System Architecture and Algorithms," The International Federation of Automatic Control, 46(10), 2013, 114-119.

C. Maple, M. Bradbury, A. T. Le, and K. Ghirardello, "A Connected and Autonomous Vehicle Reference Architecture for Attack Surface Analysis," Applied Sciences, 9(23), 2019. doi:10.3390/app9235101

G. Muller and P. van de Laar, "Researching reference architectures and their relationships with frameworks,methods,techniques, and tools" Procs. 7th Ann. Conf. on Systems Eng. research (CSER 2009).

C. Orellana, E.B.Fernandez, H. Astudillo, "A pattern for a Secure Sensor Node", PLoP 2020.

P. Pelliccione, E. Knauss, S. M. Agren, R. Heldal, C. Bergenhem, A. Vinel, and O. Brunnegard, "Beyond connected cars: A systems of systems perspective," Science of Computer Programming, 2020, 191.

C. Plappert, D. Zelle, C. Krauß, B. Lange, S. Mauthofer, J. Walter, and T. V. Pape, "A Privacy-aware Data Access System for Automotive Applications," 15th escar Europe - The World's Leading Automotive Cyber Security Conference, 2017, Berlin, Germany.

V. Romero, and E. B. Fernández, "Towards a Security Reference Architecture for Cyber- Physical Systems, "The 15th LACCEI International Multi-Conference for Engineering, Education, and Technology: "Global Partnership for Development and Engineering Education", 2017, Boca Raton, FL, United States: LACCEI.

A. Sahu, E. B. Fernandez, M. Cardei, and M. Vanhilst. 2010. A Pattern for a Sensor Node. In Proceedings of the 17th Conference on Pattern Languages of Programs (PLOP '10). Association for Computing Machinery, New York, NY, USA, Article 7, 7 pages. DOI:http://dx.doi.org/10.1145/2493288.2493295

A. C. Serban, E. Poll, and J. Visser, "A Standard Driven Software Architecture for Fully Autonomous Vehicles," 2018 IEEE International Conference on Software Architecture Companion (ICSA-C), 120-127, Seattle, WA

V. Stricker, K. Lauenroth, P. Corte, F. Gittler, S. De Panfilis, and K. Pohl, "Creating a Reference Architecture for Service-Based Systems – A Pattern-Based Approach," in Towards the Future Internet - Emerging Trends from European Research, G. Tselentis, A. Galis, A. Gavras, S. Krco, V. Lotz, E. Simperl, B. Stiller, and T. Zahariadis (Eds.) IOS Press, 2010.

U. S. Department of Transportation, "Architecture Reference for Cooperative and Intelligent Transportation,", 2019, Retrieved March 03, 2020 from https://local.iteris.com/arc-it/html/architecture/architecture.html

A. Uzunov, E.B. Fernandez, and K.Falkner, "ASE: A Comprehensive Pattern-Driven Security Methodology for Distributed Systems", Journal of Computer Standards & Interfaces , Volume 41, September 2015, 112-137, http://dx.doi.org/10.1016/j.csi.2015.02

A. M. Wyglinski, X. Huang, T. Padir, T. R. Eisenbarth, and K. Venkstasubramanian, "Security of Autonomous Systems Employing Embedded Computing and Sensors," IEEE Computer Society, 33(1), 2013, 80-86.

T. Zhang, H. Antunes, and S. Aggarwal, "Defending Connected Vehicles Against Maiware: Challenges and Solution Framework," IEEE Internet of Things Journal, 1(1), 2014, 10-21.