Functional Layered Architectures and Control Solutions in Internet of Vehicles – Comparison

Eugen Borcoci
University POLITEHNICA of Bucharest
Bucharest, Romania
e-mail: eugen.borcoci@elcom.pub.ro

Marius Constantin Vochin
University POLITEHNICA of Bucharest
Bucharest, Romania
e-mail: marius.vochin@upb.ro

Serban Georgica Obreja
University POLITEHNICA of Bucharest
Bucharest, Romania
e-mail: serban@radio.pub.ro

***Abstract* — The Internet of Vehicles is a novel development trend in vehicular networking. Its driving factor is, on one part, the high growth of the vehicles number, including the intelligent ones and the need to solve numerous problems encountered in transportation systems related to safety, traffic management, information and entertainment services, autonomic vehicles challenges and so on. Internet of Vehicles extends the capabilities of the traditional Intelligent Transport System technologies but also takes benefit from new technologies used in Future Internet. It is considered by many authors as a sub-domain of Future Internet and specifically of Internet of Things. Internet of Vehicles will integrate the previous Vehicular Networks and also functionalities already developed in ITS. However, there is no unique definition of what Internet of Vehicles exactly is; some concepts and architectural aspects are still open research issues. This paper is not an exhaustive survey; it attempts a comparative critical analysis of several functional architectures and systems proposed for Internet of Vehicles. Recent approaches Fog/Edge computing – based systems and Software Defined Networking are also considered. An enriched SDN/Fog based architecture is proposed.**

Keywords — Internet of Vehicles; V(A)NET; Fog computing; Edge computing; Software Defined Networking; Network Function Virtualization.

I. INTRODUCTION

This paper is an extended version of the work [1], dedicated to a comparative analysis of some relevant functional architectures recently proposed for the *Internet of vehicles* (IoV). The aim here is not to detail certain functions or services, but to evaluate several variants of structured layering of functions and possible separation of functions among several architectural planes.

Vehicular communications, networks and many associated services have been intensively studied, designed, standardized and also implemented in the last two decades. The driving force has been and still is, the significant growth of the vehicles number all over the world, together with many problems related to transportation, but also due to the market needs of new services available in vehicular environment. The umbrella and framework for such developments is the *Intelligent Transport System* (ITS) [2].

Complementary support networking technologies have been developed in this area including the lower layers (physical and data link layers for wireless access) *Dedicated Short-Range Communications* (DSRC) and also higher functional layers *Wireless Access in Vehicular Environments* (WAVE) [3]. The IEEE 802.11a/p and respectively IEEE 1609 represent a mature set of standards for DSRC/WAVE networks. For wide area and high-speed mobility, another solution for wireless access of vehicles is based on 4G, *Long Term Evolution* (LTE) technology and recently on LTE-A (LTE-Advanced). Experiments have shown that the vehicles can operate with the speed of ~150 km/h. An alternative to LTE is WiMAX (World-wide interoperability for Microwave Access).

Vehicular Ad Hoc Networks (VANET) [4] have been defined to support basic vehicular communications types: *vehicle to vehicle (V2V), vehicle to road (V2R),* or *vehicle to Infrastructure (V2I)* in uni-directional or bi-directional communications (note that, some authors include V2R into V2I type). The basic VANET functional components are the *On-Board-Unit* (OBU), installed into the vehicles and *Road-Side–Unit* (RSU) placed on the roads. The RSUs communicates with vehicles, can inter-communicate and also could be linked to external networks like Internet. The main applications and services of VANET have been oriented to safety and traffic management use-cases.

The VANETs have several limitations related to their pure ad hoc network architecture (in V2V case), unreliable Internet service, incompatibility with personal devices, non-cooperation with cloud computing, low accuracy of the services, operational network dependency and restricted areas of applications and services. Therefore, extending the VANET architecture is considered today as a strong need and an opportunity.

Recently, *Internet of Vehicles* (IoV) concepts and architectures have been proposed as a significant enhancement in vehicular communication area. IoV could be seen as a global span of a vehicle network [5-9]. On the other part, IoV is considered as a special case of *Internet of Things* [10] [11], where the "things" are either vehicles or their subsystems. The IoV will connect the vehicles and RSUs through different *Wireless/Radio Access Technologies* (WAT/RAT), while traditional Internet and other heterogeneous networks will be used for wide area. In terms of services, IoV has as objectives to include the traditional VANET services but also will be open for development of novel ones, e.g., vehicle traffic management in urban or country areas, automobile production, repair and vehicle insurance, road infrastructure construction and repair, logistics and transportation, etc.

The IoV can be strongly supported by recent technologies like centralized *Cloud Computing* (CC) combined with *Fog* or *Edge Computing* [11] [12]; in comparison with CC, the Fog/Edge can offer for IoV a better time response, more flexibility and higher degree of functional distribution, context awareness, reduction in the amount of data exchanged between a cloud data center and a vehicle. All these features are more appropriate for vehicular world in comparison to centralized cloud computing approach.

In terms of management and control, *Software-defined networking* (SDN) technology [13] can offer to IoV its centralized up-to-date logical view upon the network, programmability, facilitating a flexible network management and on-the-fly modification of the network elements behavior.

Network Function Virtualization (NFV) [14] can add flexibility by virtualizing many network functions and deploying them into software packages. Dedicated *Virtualized Network Functions* (VNF) can be defined, then dynamically created/used/destroyed, assembled and chained to implement legacy or novel services. Challenges and open research issues exist, related to NFV and SDN cooperation and their adaptation to the vehicular networks requirements concerning high mobility, distributed character, aiming finally to realize new flexible and powerful IoV architectures and systems.

The large communities of users/terminal devices in IoV need powerful and scalable *Radio Access Technologies* (RAT). The 4G and the emergent 5G technologies, based on cloud computing architectures (*Cloud Radio Access Network*- CRAN) are significant candidates for constructing the IoV access infrastructure [15].

Despite IoV promises high capabilities, there still exist many challenges, both in conceptual and architectural aspects and also from implementation and deployment points of view. Many IoV advanced features and integration with the above technologies (CC, Fog/Edge, SDN, NFV) are still open research issues.

This paper attempts a comparative critical study of several functional layered architectures proposed for IoV, including recent ones based on Fog/Edge computing and Software defined networking (SDN) - control. An enriched functional architecture with Fog computing and SDN control is proposed in the paper. Other candidate support technologies for IoV, like Mobile Edge Computing are shortly discussed. The Sections III and V contains the main additional contributions w.r.t the original work [1].

The paper is organized as follows. Section II is a short overview of related work on IoT layered architecture. Section III exposes a comparative presentation of some IoV generic layered functional architectures. Section IV revisits the SDN-based architectures of IoV. Section V proposes a Fog-SDN oriented, enriched integrated architecture. Section VI presents a mapping example of the generic IoV architecture on Mobile Edge Computing (MEC) technology. Section VII draws some conclusions and exposes future work.

II. INTERNET OF THINGS LAYERED ARCHITECTURES

IoV is frequently seen as a part of the more general Internet of Things (IoT), so it is of interest to compare how the IoV architectures are generally consistent with previously proposed IoT architectures.

Among several architectural overviews and stacks suggested for IoT, Al-Fuqaha et al. [10] present an interesting IoT overview. They identify several IoT elements, i.e., *identification, sensing, communication, computation, services and semantics.* Several variants of IoT layered architectures are presented, where the most comprehensive has 5-layers:

- *Business (BL)*- highest layer
- *Application (AL)*
- *Service Management (SML)*
- *Object Abstraction (OAL)*
- *Objects (perception) (OL)*- lowest layer

If compared with the classical TCP/IP architecture, the above layers are defined in a more general way, but the layering principles are still preserved, in the sense that a given layer offers a set of services to the upper layer.

The *Object (perception)* layer (lowest) represents the IoT physical sensors and actuators, performing functionalities such as querying location, temperature, weight, motion, vibration, acceleration, humidity, etc. The digitized data are transferred to the OAL through secure channels.

The *Object Abstraction* layer transfers abstracted data to the *Service Management layer* through secure channels. Traditional Layer 2 networking transfer functions are included here, based on technologies like RFID, GSM, 3G, 4G, UMTS, WiFi, Bluetooth Low Energy, infrared, ZigBee, etc. Additionally, cloud computing capabilities are offered, and data management processes are handled at this layer.

The *Service Management* layer plays a middleware role, by pairing a service with its requester based on addresses and names. The SML supports IoT application programmers to work with abstracted heterogeneous objects. It also processes received data, takes decisions, and delivers the required services over the network wire protocols.

The *Application* layer provides to the customers the requested services (with appropriate quality). The AL covers different vertical markets (e.g., smart home, transportation, industrial automation, health care, etc.).

The *Business Layer* manages all IoT system activities and services. Using data provided by AL, it creates a business model, graphs, flowcharts, etc.; it is related to design, analysis, implementation, evaluation, monitoring and management (of the lower layers), and developing IoT system related elements. Decisions can be taken following Big Data analysis. Security features are included. Note that the architecture described above is a high-level view only; further structuring can be made and mapping on various existing protocols.

The work of Khan et al. [16] also proposes a five-layer architecture for IoT, which is similar to the previous one.

1) *Perception/Device Layer (PL):* consists of the physical objects and sensors (RFID, 2D-barcode, etc.) and basically deals with the identification and collection of objects specific information by the sensor devices. The collected information is then passed to Network layer for its secure transmission to the information processing system.

2) *Network Layer (NL):* securely transfers the information from sensor devices to the information processing system. The transmission technologies can be 3G, UMTS, Wifi, Bluetooth, infrared, ZigBee, etc.

3) *Middleware Layer (ML):* is responsible for the service management and has links to the database. It processes information received from NL and store it in the database. It takes automatic decision based on the results.

4) *Application Layer (AL):* provides global management of the vertical applications based on the objects information processed in the Middleware layer.

5) *Business Layer:* manages the overall IoT system including the applications and services. It builds business models, graphs, flowcharts etc based on the data received from AL.

III. IoV Generic Layered Architectures

Several IoV architectures have been recently proposed and discussed. A short critical overview and comparison are exposed below.

Bonomi et al. [5] proposed a four - layered architecture for connected vehicles and transportation. The layers are also called *"IoT key verticals"*, suggesting that a given layer includes not only classical layer functions (i.e., L1, L2,) but rather groups of functions, which could be mapped on one or more classical layers. Also, the four layers are rather corresponding to different geo-locations of the subsystems (vehicles, networking infrastructure, cloud data centers, etc.).

The bottom layer (*end points*) represents the vehicles, plus their communication protocols (basically for V2V communication, using the IEEE 802.11a/p).

The layer two (*infrastructure*), represents communication technologies to interconnect the IoV actors (via WiFi, 802.11p, 3G/4G, etc.).

The third layer (*operation*) performs management actions; it verifies and ensures compliance with all applicable policies, to regulate the information management and flow.

The fourth layer is called *services*/*cloud* (public, private or enterprise) based on a defined profile coupled with the possibility of receiving services (voice, enterprise video and data) on demand. Note that this architectural view is a mixed one and does not clearly separate the sets of functions of various levels.

Note the partial similarity of the above architecture to those described in Section II. However, the cloud layer in the IoV is considered as the top layer in Bonomi's case, including the applications and business functions of the previous IoT architectures.

Kayvartya et al. [6] have proposed a comprehensive IoV five-layer architecture, to support an enriched set of vehicular communications, in addition to traditional V2V, V2R/V2I, i.e., *Vehicle-to-Personal* devices (V2P) and *Vehicle-to-Sensors* (V2S). Each IoV communication type can be enabled using a different WAT, e.g., IEEE WAVE for V2V and V2R, Wi-Fi and 4G/LTE for V2I, CarPlay/NCF (*Near Field Communications*) for V2P and WiFi for V2S. The system includes vehicles and *Road Side Units* (RSU), but also other communication devices. Embedding such a large range of devices makes IoV more complex, (compared to VANET), but more powerful and market oriented.

This architecture goes further than only proposing a generic overall model; separation in three architectural planes is defined: *management*, *operation* and *security*. Such a split is important because it allows later to map various existing protocols and functions (e.g., taken from ITS) to be more easily mapped on architectural layers. The network model is composed of three functional entities: *client, connection* and *cloud.* The layers are (see Figure 1): *perception, coordination, artificial intelligence, application* and *business.*

The *perception* layer (PL) functions generally include those of the traditional physical layer but have also some additional functions related to sensing and actuating actions. The PL is instantiated by *sensors* and *actuators* attached to vehicles, RSUs, smart-phones and other personal devices. The PL main task is to gather information on vehicle, traffic environment and devices (including movement–related parameters).

The *coordination* layer (CL) represents a virtual universal network coordination entity for heterogeneous network technologies (WAVE, Wi-Fi, 4G/LTE, etc.). It creates a unified communication structure for the terminal devices.

The *artificial intelligence* layer (AIL) is represented by a generic virtual cloud infrastructure, working as an information processing and management centre. It stores, processes and analyses the information received from the lower layer and then takes decisions. Its major components are: *Vehicular Cloud Computing (VCC), Big Data Analysis (BDA)* and *Expert System.* The AIL should meet the requirement of applications and services working on top of it.

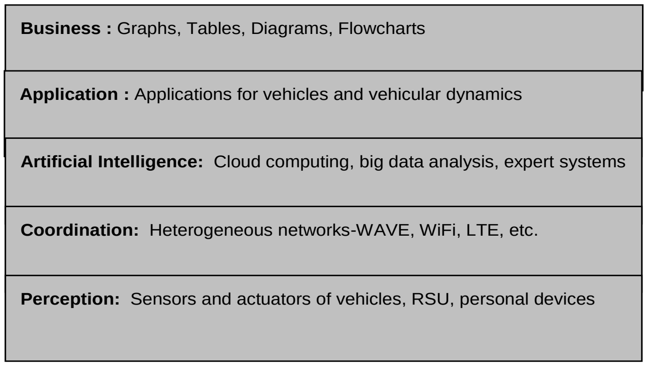


Figure 1. Five-layer IoV architecture (adapted from [6]).

The *application* layer (AL) contains smart applications (e.g., for traffic safety and efficiency, multimedia-based infotainment and web-based utility). The AL includes safety and efficiency applications (VANET legacy) and provides smart services to End Users (EU) based on intelligent analysis done by AIL. The AL efficiently discovers the services provided by AIL and manage their combinations. It also provides EU application usage data to the business layer. Currently, it is recognized that these smart applications constitute a major driving force to further develop IoV.

The *business* layer (BL) includes IoV operational management functions, basically related to business aspects: to foresight strategies for the development of business models based on the application usage data and statistical analysis of the data; analysis tools including graphs, flowcharts, comparison tables, use case diagrams, etc.; decision making - related to economic investment and usage of resources; pricing, overall budget preparation for operation and management; aggregate data management.

The architecture is split in three parallel planes: *operation*, *management and security*. The work [6] also proposed a possible mapping between the five layers and different protocols already developed in vehicular communications by ITS, VANET, IEEE, etc. The *operation* plane basically contains traditional *data* plane functions but still has some control and management role.

At *perception* layer, current network technologies can be used for access in ITS and VANET (see Figure 2).

The *coordination* layer includes not only TCP/IP transport and network protocols but also different solutions (with no IP usage). Examples are: IEEE 1609.4 along with a *Global Handoff Manager* (GHM-open research) and other protocols proposed at network layer in projects like CALM, WAVE. For instance, in the stack there exist WSMP - Short Message Protocol and FAST -Fast Application and Communication Enabler.

In the *Artificial Intelligence* layer, cloud capabilities are seen as major contributors, working on top of lower sub-layer: CALM Service Layer (CALM-SL) and WAVE-1609.6 service related protocols. The upper sub-layer consists in Vehicular Cloud Computing (VCC) and Big Data Analysis (BDA) related protocols. They can offer cloud services of type "*X as a Service*": Storage (STaaS), Infrastructure (INaaS), Network (NaaS), Cooperation (CaaS), Entertainment (ENaaS), Gateway (GaaS), Picture (PICaaS) and Computing (COMaaS).

Still further research work is necessary, given the current unavailability of enough suitable protocols for VCC and BDA. Another open issue is that VANETs projects, generally, do not have clear definitions of the upper sub-layer, while some IoT projects are recently working towards these.

The *Application* layer includes two sets of applications: *Smart Safety and Efficiency* (SSE) and *Smart Business Oriented* (SBO). The current WAVE resource handler protocol 1609.1 can be used on the top of these applications, to manage the resources among smart applications. The *Business Layer* (BL) in [6] proposes various business models like Insurance (INS), Sale (SAL), Service (SER) and Advertisement (ADV). The set of these functionalities could be further enriched in the future.

The architecture has the merit to integrate in the management and security planes some existing functional blocks and protocols (see Figure 2), already developed in WAVE (P1609.x), CALM and C2C projects.

However, the mentioned 5-layer architecture does not touch some important and recent aspects in developing IoV architecture, e.g., how to distribute computation intelligence between a central cloud and fog/edge units (placed at the network edge) while cloud-fog/edge combination seem to be an efficient and attractive solution for a distributed system like IoV. Also, SDN-like control and NFV implementation possibilities are not discussed in this architecture.

F.Yang et al. work [7] proposes a more comprehensive view on IoV architecture, based on functional requirements and proposed goals, by considering the *driver-vehicle-environment* coordination. IoV is defined as an open converged network system (controllable, manageable, operational, and trustable) based on multi-human, multi-machine, multi-vehicle, and environment coordination. It senses, recognizes, transmits, and computes the large-scale complex static/dynamic information of human, vehicle, network communication and road traffic infrastructure, using advanced ICT technology.

The architecture [7] defines four layers: the *environment sensing and control* layer, *network access and transport* layer, *coordinative computing control* layer, and *application* layer (see Figure 3). The work also summarizes the core technologies of each layer. In the *environment sensing and control* layer, vehicle control and environment sensing technologies are introduced. The network access and transport layer use the current technologies available for vehicle access and communication (access and core networks).

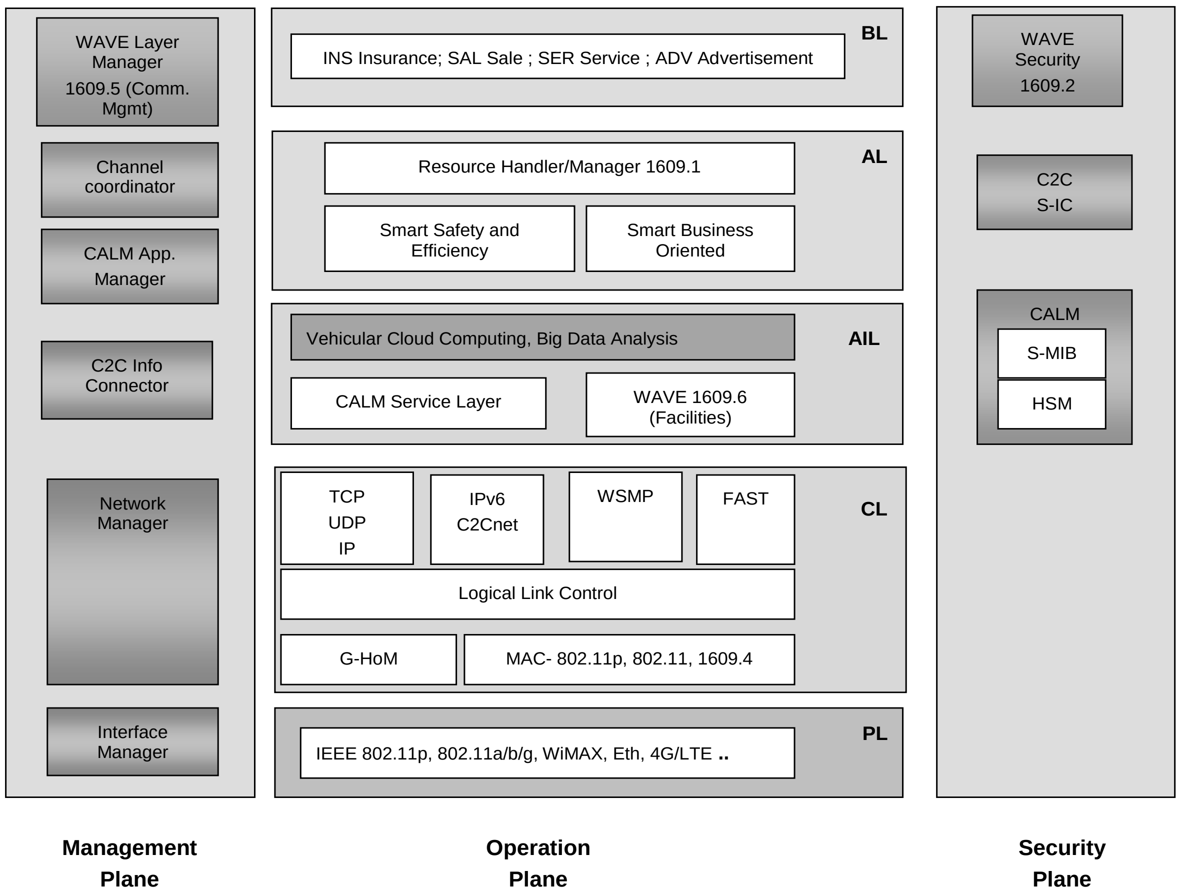


Figure 2. Five-layer IoV architecture mapped on particular protocols (adapted from [6])

PL Perception layer; CL Coordination layer; AIL Artificial Intelligence layer; AL Application layer; BL Business layer; C2C Car to Car; CALM Communication Architecture for Land Mobile; DSRC Dedicated Short Range Communication; WAVE Wireless Access in Vehicular Environment; FAST Fast Application and Communication Enabler; LLC Logical Link Control; G-HoM Global Handoff Manager; WSMP WAVE Short Messages Protocol; BDA Big Data Analysis; VCC Vehicular Cloud Computing; SSE Smart Safety and Efficiency; SBO Smart Business Oriented; INS Insurance; SAL Sale ; SER Service ; ADV Advertisement; HSM Hardware Security Manager; S-IC Security Information Connector; S-MIB Security Management Information Base

A special layer (differently defined w.r.t. other IoV architectural proposals) is introduced, i.e., the *coordinative computing control* layer. Here, the coordination among *human-vehicle-environment* in IoV is considered as a main goal. The application layer splits the services in two classes: close and open.

The coordination concept in [7] is based/divided on/in two models and, correspondingly, two objects:

a. *individual coordination model*- dealing with the capabilities of the *pair human-vehicle* and assuring coordinative computing control in the IoV environment. It solves the coordination problems between human and vehicle, and between individual object and swarm object. The in-vehicle network is involved here.

b. *swarm coordination model;* the swarm object consists of all objects of IoV except the individual object. The environment network is involved here.

The *vehicle network environment sensing and control* layer offers the basis for IoV services, including those for autonomous vehicle. The environment sensing is the recognition basis for IoV services, such as services of autonomous vehicle, intelligent traffic, and vehicle information.

From the perspective of vehicles, they sense environment information around these vehicles via autopilot system, traffic jam auxiliary system, and sensor system for achieving auxiliary driving. In terms of environment, this layer monitors and extracts various dynamic information of human, vehicles, and environment through sensing technology. It receives and executes coordinative control instructions and then feedback result to cooperative control. It contributes to the implementations of swarm sensing in swarm model.

The *network access and transport* layer mainly realize the network access, data processing, data analysis, and data transmission, remote monitoring and nodes management. It implements the inter-connection and information exchange, between entities, manages the connectivity resources and balances information load. When it is the case, it can offer a stable and quality-guaranteed information and communication transport.

The *coordination computing control layer* performs network-wide coordinative computing and control for human-vehicle-environment (data processing, resource allocation and swarm intelligence computing). This layer should include both capabilities to solve the individual model (human-vehicle) related functions and also capabilities for the multi-human and multi-vehicle coordinative computing control and service coordinative management, to support the swarm intelligence computation and various services. This layer should also provide the capability of communication coordinated management.

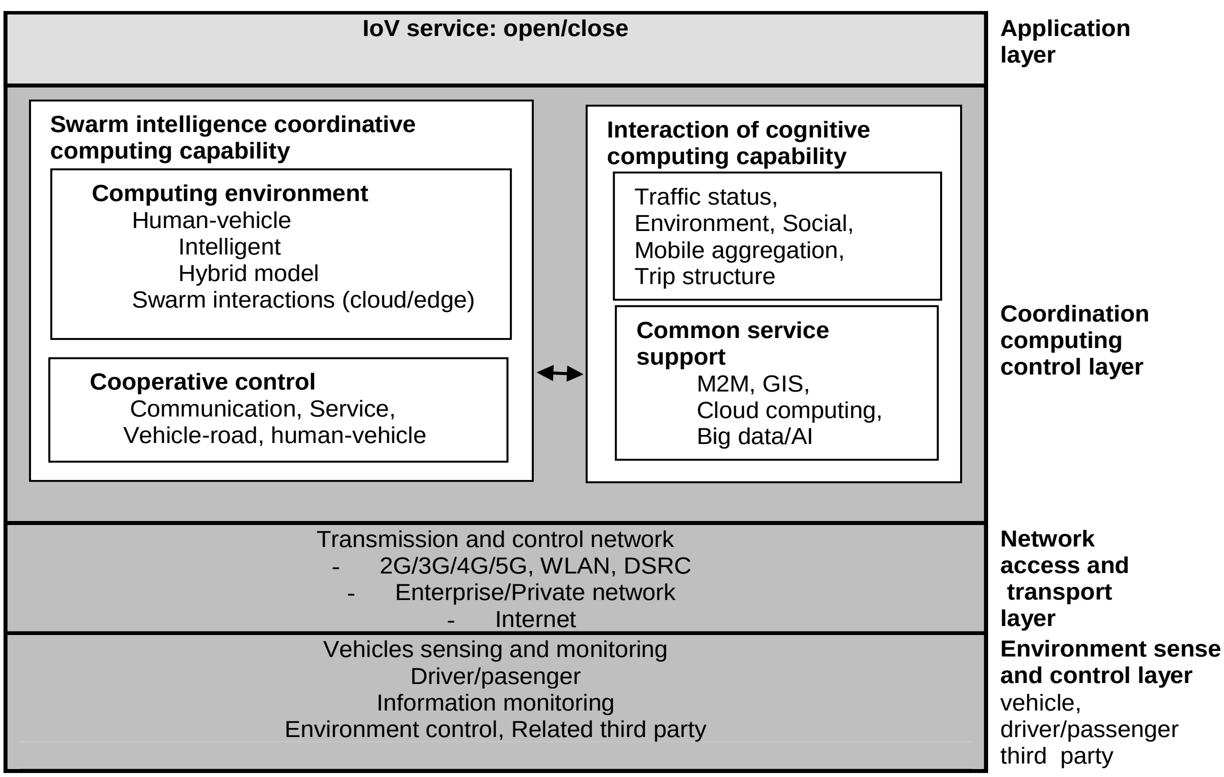
The *application* layer is defined to provide various types of services. It should be open in the sense that could support novel services and business operating modes. The application layer can be classified into closed services (related to the specific industry applications) and open services (i.e., various existing open applications, such as real-time traffic services provided by Internet service providers or to third party providers).

The architecture [7] presents (see Figure 3) in a generic way, the four layers and their internal components. *However, the criteria of splitting the entities/functions of the components included in the coordination computing layer are not very visible, e.g., between the two blocks: swarm intelligent coordinative computing and interaction of cognitive computing capabilities.*

The homogeneity of sub-layers is low in terms of their components. The separation of the overall architecture in different architectural planes is not discussed; therefore, is rather difficult to see the mapping of different already developed functions and protocols (ITS, WAVE, etc.) to the layers of this architecture. This seems to be still an open issue of this architecture.

No consideration about using technologies like SDN, NFV, Fog/edge computing (except a proposal of a virtual vehicle -VV) are mentioned. A refinement of this architecture and more precise structuring would be needed.

Contreras-Castillo et al. [8] propose a seven-layer architecture, supporting the functionalities, interactions, representations and information exchanges among all the devices inside an IoV ecosystem. The authors claim that this architecture (having more than five layers) has as objective to reduce the complexity of each layer and better standardize the interfaces and protocols used in each layer.



F

Figure 3. Four-layer IoV architecture (adapted from [7]).

The interaction model considers the following entities, which can communicate to each other: vehicle (V), person (P), personal device (PD), network infrastructure (I), sensors (S), any device (D) and roadside device (R). Consequently, the communications types might be: V2V, V2R, V2I, V2D, V2P, V2S, D2D.

The network model should support collaboration between multi-users, multi-vehicles, multi-devices (sensors, actuators, mobile devices, access points), multi-communication models (point to point, multi-point, broadcast, geo-cast) and multi-networks (wireless or wire networks with various technologies like WiFi, Bluetooth, WiMAX, 3G, 4G/LTE, etc.

The layers defined in [8] are (bottom-up list):

1 *User interaction* (lowest layer)
2 *Data acquisition*
3 *Data filtering and pre-processing*
4 *Communication*
5 *Control and management*
6 *Business* (highest layer)

An additional layer is named *Security*; however, it is actually a cross layer entity.

Note that *this "layered"- named architecture does not follow the principles of a layered stack architecture* (where each layer traditionally offers some services to the above one). For instance, the Control and management layer and Security layer seem to be rather architectural "planes "and not traditional layers; they have to interact with all other five layers.

The *User interaction* layer contains in-vehicle computing systems including:

a. information-based systems to provide information (e.g., on routes, traffic conditions, car parking availability and warning/advice regarding risks) to components of the driving environment, i.e., the vehicle or the driver;

b. control-based systems to monitor changes in driving habits and experiences and operational elements of the driving task (e.g., adaptive cruise control, speed control, lane keeping and collision avoidance).

It is stated in [8] that designing user interfaces for in-vehicle systems is still raising many new research challenges. Note also that this "layer" actually contains functions of several layers defined in other architectures (e.g., some structured in a similar way as classic TCP/IP stack).

The *Data acquisition* layer has tasks covering all three traditional architectural planes (data, control and management). Generally, it has functions similar to traditional Layer 2. It gathers data (for safety, traffic information, infotainment), from a given area of interest, from all the sources (vehicle's internal sensors, GPS, inter-vehicle communication, Wireless Sensor Networks (WSN), or devices such as cellular phones, sensors and actuators, traffic lights and road signals located on streets and highways. Intra- and inter-vehicular interactions are within the scope of this layer. Various access technologies and associated protocols are supposed to perform the tasks. For intra-vehicle communication, the proposals are: Bluetooth (2.4 GHz), ZigBee (868 MHz, 915 MHz and 2.4 GHz), Wi-Fi HaLow (Low power, long range Wi-Fi, 900 MHz), Ultra-wideband (3.1–10.6 GHz), with data rates up to 480 Mbps and coverage distances up 1000m. For inter-vehicles communication technologies can be: IEEE WAVE/DSRC with IEEE 802.11p for PHY and MAC layers and the IEEE 1609 family for upper layers; 4G/LTE (1700 and 2100 MHz).

The *Data filtering* and *pre-processing* layer is necessary, given that IoV, may generate huge amounts of data, while not all are relevant for all entities of the system. This layer analyses and filters the collected information, to avoid the dissemination of irrelevant information and therefore reduces the overall network traffic. Examples of protocols to be used in this layer are:

- *Xtensible Messaging and Presentation Protocol* (XMPP)
- *Constrained Application Protocol* (CoAP)
- *HTTP Representational State Transfer* (HTTP REST)
- *Message Queuing Telematic Transport* (MQTT)
- *Lightweight Local Automation Protocol* (LLAP).

Several data filtering approaches are referenced in [8], but novel intelligent and efficient data mining techniques are considered to be necessary.

The *Communication* layer performs both data and control function at the networking level, given the set of protocols suggested as: 6LoWPAN, IPv4, IPv6, Routing Protocol for Low Power and Lossy Networks (RPL), etc. This layer should select the best network to send the information, based on several selection parameters (e.g., congestion level, QoS level capabilities over the different available networks, information relevance, privacy and security, cost, etc.). Apparently, the traditional networking functions are split between this layer and Acquisition layer.

The *Control and management* layer is the global coordinator that manages different network service providers within the IoV environment. Its functions are: to manage the data exchange among the various services; to manage the information generated by devices: in-vehicle or around sensors, roadside infrastructure and user devices in the environment; apply different suitable policies (e.g., traffic management and engineering, packet inspection, etc.). *It is not yet clear what intra and inter-domain management tasks has this entity*. The protocols proposed for this layer are: *CALM Service Layer*, WAVE 1609.6, TR-069, *Open Mobile Alliance Device Management* (OMA-DM).

The *Business* layer processes information using various types of cloud computing infrastructures available locally and remotely. Typical functions are: storing, processing and analysing info received from the other layers; making decisions based on data statistical analysis and identifying strategies that help in applying business models based on the usage of data in applications and the statistical analysis. (tools such as graphs, flowchart, critical analysis, etc.).

The *Security* layer (despite of its name - "layer") is an architectural plane, which communicates directly with the rest of the layers. It implements security functions (data authentication, integrity, non-repudiation and confidentiality, access control, availability, etc.) to exchange data among sensors, actuators, user's devices through secure networks and service providers. The protocols envisaged are similar to those presented in Figure 2.

The work [8] proposes a split of the architecture in two planes: operational – containing six layers and the security plane, aiming to define the structure to include some of the current protocols in the different layers.

Considering the protocols proposed in [8] to be mapped on different layers it is apparent that the *Acquisition* layer is playing the role of access - given that access technology protocols are proposed there: Wi-Fi, 2G/3G/4G//LTE, Bluetooth, IEEE 1609/WAVE, IEEE 802.11p, WiMAX, etc. On the other side the *Communication* layer includes networking protocols like IPv4/IPv6RPL, ROLL, etc. Therefore, the two layers could have been merged as a *Access and Core Network* layer.

The cloud services are located at business level (as vehicular cloud computing) while we believe that a more natural placement could be as in Figure 2, i.e., under application layer.

Some mixture of "layers" and "plane" notions is apparent; there is a lack of enough orthogonality of different "layers". The architecture does not touch the integration of SDN/NFV approach.

Table I shows a comparison of the layered architectures exposed in this section.

TABLE I. LAYERED ARCHITECTURE COMPARISON

Layered Architecture	**Criteria of comparison**						
	No. of (macro) layers	***Target domain***	***Split in architectural planes***	***Mapping of protocols on architectural stack***	***Cloud computing included***	***SDN/NFV approach introduced***	***Edge/Fog computing approach introduced***
Bonomi et al. [5]	4	IoT/IoV	No	No	Yes (highest layer)	No	Yes
Kayvartya et al. [6]	5	IoV	Yes	Yes	Yes (middle layer)	No	No
F.Yang et al. [7]	4	IoV	No	No	Yes (middle layer)	No	Edge-only summary
Contreras-Castillo et al. [8]	6+1	IoV	Partial	Yes	Yes (highest layer)	No	No

IV. SDN CONTROLLED IOV ARCHITECTURES

Recent works emphasize the benefit of using novel technologies like SDN, NFV, cloud/fog/edge in the context of IoV. This section shortly presents samples of related work dedicated to VANET/IoV with SDN control.

Y.Lu et al. [17] apply SDN control to VANET, to get more flexibility, programmability and support for new services. The architectural components are:

- SDN controller
- SDN wireless nodes and
- SDN-enabled RSUs.

The SDN controller is a single entity performing the overall control of the system. The SDN wireless nodes are vehicles, considered as architectural Data plane elements (equivalent to SDN - forwarders). The SDN RSUs are also treated as Data plane elements, but they are stationary. The benefits of the approach are proved by simulation, while considering some specific use cases (e.g., routing). *However, a complete layered functional IoV architecture is not discussed.*

K.Zeng et al. [18] propose an IoV architecture called *software-defined heterogeneous vehicular network* (SERVICE), based on Cloud-RAN technology [15], able to support the dynamic nature of heterogeneous VANET functions and various applications. A multi-layer Cloud-RAN multi-domain is introduced, where resources can be exploited as needed for vehicle users. The system is hierarchically organized (three levels of clouds are defined: remote, local and micro clouds) and virtualization techniques (offering flexibility) are considered for implementation. The high-level design of the soft-defined HetVNET is presented. The SDN control is organized on two levels (one primary controller and several secondary controllers; each one of the latter controls a given service area). *A definition of a complete layered functional IoV architecture is not in the paper scope.*

A Fog-SDN architecture called FSDN is proposed for advanced VANET by Truong et al. [12], for V2V, V2I and Vehicle-to-Base Station communications. The Fog computing brings more capabilities for delay-sensitive and location-aware services. The SDN components (hierarchically top-down listed) are:

- *SDN Controller* (it controls the overall network behavior via OpenFlow –interfaces; it also performs Orchestration and Resource Management activities for the Fog nodes);
- *SDN RSU Controller* (RSUC) (controlled by the central SDN controller; each RSUC controls a cluster

of RSUs connected to it through broadband connections. The RSUC can forward data, and store local road system information or perform emergency services. From Fog perspective RSUCs are fog devices);

- *SDN RSU* (it is also a Fog device);
- *SDN Wireless Nodes* (vehicles acting as end-users and forwarding elements, equipped with OBU);

The system also contains *Cellular Base Station* (BS) performing traditional functions (they are also SDN-controlled via OpenFlow protocol and can also offer Fog services). *This study does not discuss a full functional layered IoV architecture*.

Kai et al. [19] present an overview of Fog–SDN solution for VANET and discuss several scenarios and issues. It is shown that a mixed architecture Fog-SDN (similar to that proposed in [12]) can be powerful and flexible enough, to serve future needs of IoV. *Again, we note that this study does not discuss a full functional layered IoV architecture.*

Chen et al. [20] discuss an IoV architecture and solutions based on SDN control. An SDN switched network is considered as a core network, controlled by SDN controllers. The vehicles are placed at the edges connected to the core via wireless data and control paths. The architectural planes are similar to those defined in SDN: data plane, control plane and application plane. To these a knowledge plane is added. *However, a full functional layered architecture with mapping of different protocols on this architecture is not discussed. Also, a fog/edge approach is missing.*

V. A SDN-Fog Enabled IoV Functional Architecture

This section proposes a layered functional IoV architecture of a heterogeneous network including SDN control and Fog computing capabilities. We propose a possible infrastructure (Figure 4), *which could be a horizontal extension of that proposed in* [12] for large network configurations (based on definition of regional service areas). Also, an enrichment of the five-layered architecture of [6] is proposed to introduce the functions of SDN control and also Fog computing.

The Data plane includes: mobile units (vehicles) equipped with OBUs; advanced RSUs, which could have enough resources (computing, storage) as to play also Fog node role (F-RSU) or could be regular RSU like in traditional VANETs; base stations (BS) of type WiMAX/3G/4G-LTE. Note that the BSes could also have fog-node capabilities (F-BS notation is used for such cases) A fixed network (partial mesh) can interconnect the RSUs.

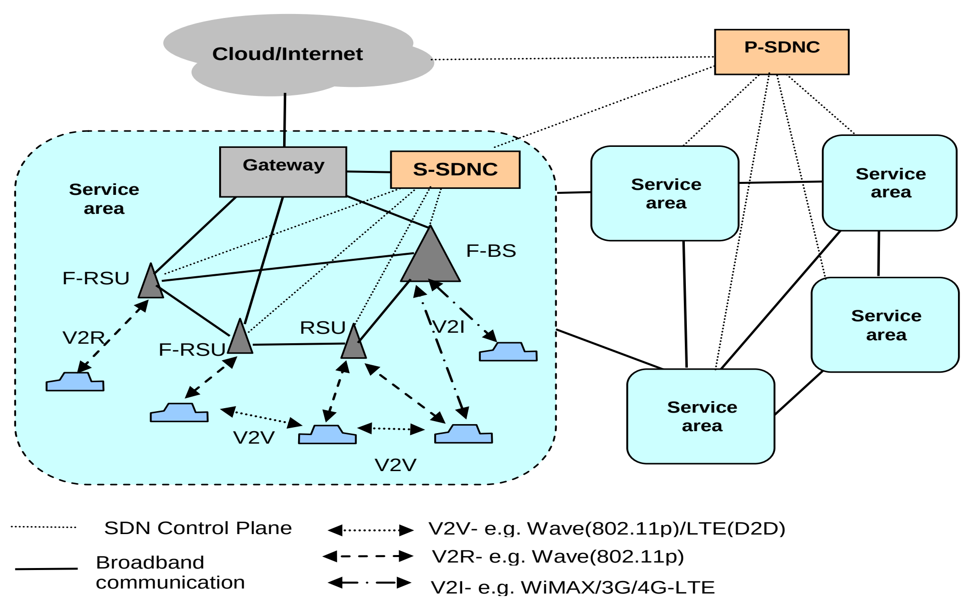


Figure 4. Generic IoV system architecture proposed in this paper.
F-BS - Fog-capable Base Station; F-RSU Fog-capable Remote Side Unit; P-SDNC- Primary SDN Controller;
S-SDNC Secondary-SDN Controller; D2D- device to device communication

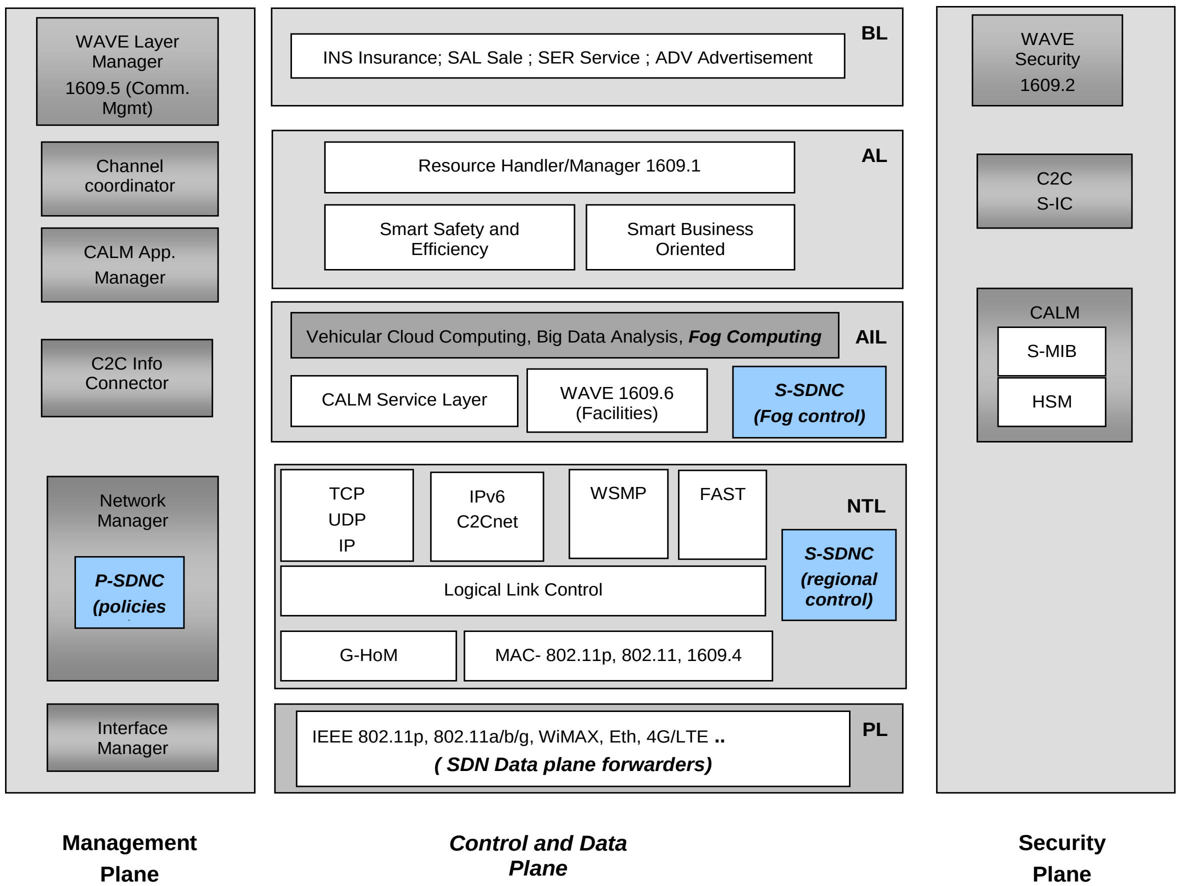


Figure 5. Functional IoV five-layer architecture enriched with SDN control and Fog computing capabilities (extended architecture of [6])

The SDN Data plane contains the forwarding nodes and can be geographically organized in several *service areas*.

The SDN Control plane is organized on two hierarchical levels: primary SDN controller (P-SDNC) controlling the overall behavior of the network and secondary controllers (S-SDNC), one for each service area. The S-SDNC can also contain the resource management functions of the Fog infrastructure. The P-SDNC is logically connected to each S-SDNC via the Control plane overlay-type or physical links. The SDN south interfaces between the controllers and the lower level can be supported by OpenFlow protocol or some other similar protocol. This infrastructure is enough general as to be considered as a candidate or IoV.

Figure 5 shows a proposal to enrich the layered functional architecture introduced by Kaiwartya et al. in [6], by adding SDN and Fog functionalities, supposing that the infrastructure is that of Figure 4. The second layer of the architecture is renamed in Network and Transport Layer (NTL), showing in a more explicit way the role of this layer. The Operation Plane is renamed in Control and Data Plane.

The functionalities of the P-SDNC can be embedded in the management plane, given that its role is to govern the overall network behavior (e.g., some overall policies can be coordinated by this module). The regional SDN control is placed naturally at NTL level as to control the SDN forwarders and also the functions of Fog nodes located in the access area. Additionally, S-SDNC functions can be included in the AIL, to serve this layer needs in terms of Fog AI resource control.

In a complete system the cloud-based services could be split between centralized Cloud computing and Fog nodes. How to manage this in an efficient way is not solved in this paper; it is for further study.

Note that a complete architectural definition for the above proposal is also for further study. While it can be mapped onto five-layer architecture described in Section III, no details are developed here on how the virtualization will be managed and how the off-loading actions are performed in order to preserve the service continuity when the vehicles are moving.

VI. MOBILE EDGE COMPUTING - BASED IOV ARCHITECTURE

Mobile Edge Computing (MEC) technology is an edge-oriented computing technology exposing all advantages of edge computing: low latency/response time, high bandwidth, location and context awareness, reduction in amount of data transferred from a terminal device to a centralized cloud data center and back, reduced round-trip time, etc. [21] [22]. The MEC cloud computing resources and storage spaces are placed at the edge of the vehicular access network (usually in Radio Access Netwok - RAN) and are in close proximity to the mobile terminal.

MEC is a distributed computing environment where applications can benefit from real-time radio and network information and can offer a personalized and contextualized experience to the mobile subscriber. The mobile-broadband experience is more responsive and opens up new monetization opportunities. This creates an ecosystem where new services are developed in and around the Base Station.

The key element is MEC application server, usually integrated in RAN, which can provide computing resources, storage capacity, connectivity, and access to user traffic and radio and network information. MEC offers an open radio network edge platform, supporting multi-service and multi-tenancy. Authorized third-parties may also to make use of the storage and processing capabilities, introducing new businesses on-demand and in a flexible manner.

The main standardization organization involved in MEEC is ETSI, which established in 2014 the Mobile Edge Computing Industry Specification Group. Recently, in 2017, the name MEC has been changed into Multi-access Edge Computing [23] - to better reflect non-cellular operator's requirements and fixed access case.

The general MEC architecture is presented in Figure 6 [22]. The mobile edge host level is the main MEC sub-system consisting of two main parts: the mobile edge host and the mobile edge host level management. The mobile edge host provides the virtualization infrastructure (based on Network Function Virtualisation Infrastructure –NFVI- coming from ETSI Network Function Virtualization -NFV framework) and the mobile edge platform, supporting the execution of mobile edge applications.

MEC has a good perspective as a supportive technology for vehicular communication (V2V, V2I, etc.) and IoV [24] [25]. Vehicles connected to the distributed edges may send/receive information from other vehicles or through the network almost in real-time. The mobility of vehicles is naturally supported by the RAN. However, not many publications in this area exist yet.

K.Zhang, et al., [24] developed a MEC-based model of a vehicular network. Their architecture comprises several levels: *Virtual Computation Resource Pool*- incorporating the network and cloud resources outside the MEC; MEC level – implemented as MEC servers placed in the RAN; RSUs units placed on the roads and mobile unites (vehicles). *This study does not offer a detailed layered architecture, neither aspect of implementation using SDN, NFV technologies, except the implicit use of NFV management and NFVI to realize the MEC architectural stack.*

The main focus of this work is on the computation off-loading process, to preserve the service continuity in the MEC environment. Due to their high mobility, vehicles in transit may pass through several RSUs and MEC servers during the task-off-loading process, and they can off-load their computation task to any MEC servers that they can access. Two methods are possible: selection of the target MEC servers or selecting (for a while) of a new path from the mobile vehicle to the same MEC server (keeping as much as possible the same serving MEC server in order to avoid too frequent moving of virtual machines).

J.Liu et al. [25] propose an SDN-enabled network architecture assisted by MEC, while integrating different types of access technologies.

The architectural components of the overall system are (top-down hierarchical list): Remote Data Center; Backbone network, Regions (MEC server + SDN controller, BS and mobiles organized in VANETs). The MEC servers can inter-communicate via a mesh of fixed network links.

The layered architecture [25] is less elaborated than that proposed in [6]. This one is SDN-like comprising three planes (Data, Control and Application) each including typical functions:

Data Plane (DPl): SDN- "switches" (VANET, BS, Ethernet); lower layer technologies (IEEE 802.11p, LTE/5G, Wire NIC, etc.).

Control Plane (CPl)

- lower sub-layer: Position/Channel sensing, Flow table management, Forwarding strategy;
- upper sublayer: Trajectory prediction, Interface sensing, Radio Resource control, Traffic redirection.

Application Plane (APl) (in the SDN semantics): Topology management, Resource Management, Traffic Offload, SDN controller.

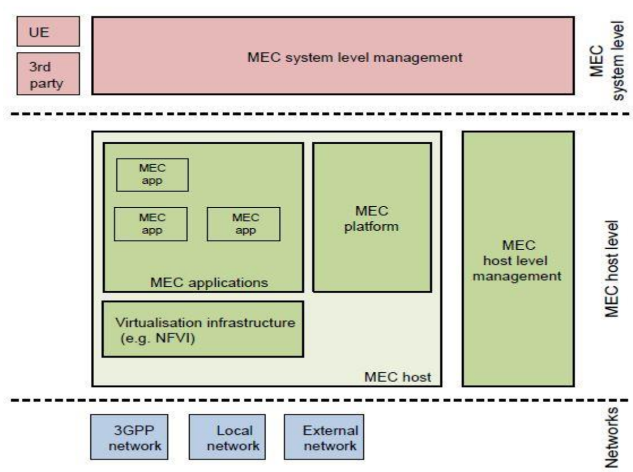


Figure 6. Simplified MEC architecture (ETSI)

The interface between CPl and DPl is based on extended OpenFlow or other similar protocol. *The limitation of this architectural proposal is that no mapping on SDN/NFV/ and fog/edge approach is discussed.*

VII. Conclusions and Future Work

This paper presented a comparative critical view of several IoV architectures proposed in the literature, focused on functional layering aspects.

Among several proposals, we selected a five-layer multiple-plane architecture, considering this model as a good and orthogonal approach, which consistently include the major IoV functionalities and is giving the possibility to clearly define interfaces between layers and architectural planes. Another advantage is that the architecture is consistent with IoT architectural vision. Several examples based on SDN/Fog/Edge approaches are comparatively discussed, mainly from the point of view of layering the architecture and map different protocols on it.

In Section V, a modified Fog-SDN based IoV infrastructure is proposed by the authors, where the associated layered architecture is enriched by considering the additional Fog-based approach and SDN distributed control. This work could be a contribution towards an IoV reference architecture.

Section VI shortly present some MEC-based IoV systems, as an alternative to Fog-based approach. The comparative study of MEC/Fog alternatives also are topics for further work.

Future work should be done to allocate and map different functions of the general functional layered architecture to specific entities of a complete IoV system. This should be done based on their different roles and placement: terminals (vehicles), RSUs, Fog/Edge Nodes, BS, core network, cloud data centers, etc. The virtualization challenges and their impact on the architecture are not yet discussed in this study. This is also subject for further work.

References

[1] E. Borcoci, S.G. Obreja, M. Vochin, “Internet of Vehicles Functional Architectures - Comparative Critical Study”, The Ninth International Conference on Advances in Future Internet, AFIN 2017, September 10 - 14, 2017 - Rome, Italy, http://www.iaria.org/conferences2017/ProgramAFIN17.html, [Retrieved: December, 2017].

[2] ETSI EN 302 665 V1.1.0 European Standard Telecommunications series, "Intelligent Transport Systems (ITS);Communications Architecture" (2010-07).

[3] Y. Li, “An Overview of the DSRC/WAVE Technology”, https://www.researchgate.net/publication/288643779, 2012, [Retrieved: May, 2017].

[4] S. Sultan, M. Moath Al-Doori, A.H. Al-Bayatti, and H. Zedan “A comprehensive survey on vehicular Ad Hoc Network”, J.of Network and Computer Applications, Jan. 2014, https://www.researchgate.net/publication/259520963, [Retrieved: December, 2016].

[5] F. Bonomi, “The smart and connected vehicle and the Internet of Things”, San José, CA: WSTS, 2013, https://tf.nist.gov/seminars/WSTS/PDFs/1-0_Cisco_FBonomi_ConnectedVehicles.pdf, [Retrieved: December, 2016].

[6] O. Kaiwartya, A.H. Abdullah, Y. Cao, A. Altameem, and M. Prasad, “Internet of Vehicles: Motivation, Layered Architecture, Network Model, Challenges, and Future Aspects” IEEE Access, Special Section on Future Networks, Architectures, Protocols and Applications, Vol. 4, pp.5536-5372, September 2016.

[7] F.Yang, J.Li, T. Lei, S. Wang, “Architecture and key technologies for Internet of Vehicles: a survey”, Journal of Communications and Information Networks, Vol.2, No.2, Jun. 2017, DOI: 10.1007/s41650-017-0018-6

[8] J.C. Contreras-Castillo, et al., “A seven-layered model architecture for Internet of Vehicles”,Journal of Information and Telecommunications , Vol. 1, No. 1, pp. 4–22, 2017.

[9] Y. Fangchun, W. Shangguang, L. Jinglin, L. Zhihan, and S. Qibo, “An overview of Internet of Vehicles”, China Commun., vol. 11, no. 10, pp. 115, October 2014.

[10] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications”, IEEE Communications Surveys &Tutorials Vol. 17, No. 4, pp.2347-2376, 2015.

[11] F. Bonomi, R. Milito, J. Zhu, and Sateesh Addepalli, “Fog Computing and Its Role in the Internet of Things”, August 2012, http://conferences.sigcomm.org/sigcomm/2012/paper/mcc/p13.pdf, [Retrieved: January, 2017].

[12] N.N. Truong, G.M. Lee, and Y. Ghamri-Doudane, “Software defined networking-based vehicular ad hoc network with fog Computing”, Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM’15), May 2015, Ottawa, Canada. Piscataway, NJ, USA: IEEE, , pp. 1202–1207, 2015.

[13] B. N. Astuto, M. Mendonca, X.N. Nguyen, K. Obraczka, and T. Turletti, “A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks”, Communications Surveys and Tutorials, IEEE Communications Society, (IEEE), 16 (3), pp. 1617 – 1634, 2014.

[14] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, “Network Function Virtualisation: Challenges and Opportunities for Innovations”. IEEE Communications Magazine, pp. 90-97, February 2015.

[15] M. Peng, Y. Li, J. Jiang, J. Li, and C. Wang, “Heterogeneous cloud radio access networks: a new perspective for enhancing spectral and energy efficiencies”, IEEE Wireless Communications, pp.126-135, December 2014.

[16] R. Khan, R. Zaheer, S. Khan, “Future Internet: The Internet of Things Architecture, Possible Applications, and Key Challenges”, 10th International Conference, on Frontiers of Information Technology, 2012, https://www.academia.edu/35984783/Future_Internet_The_Internet_of_Things_Architecture_Possible_Applications_and_Key_Challenges?auto=download [Retrieved : Dec.2017]

[17] Y. Lu, M. Gerla, R. Gomes, and E. Cerqueira, “Towards software-defined VANET: Architecture and services”, MedHocNet.2014.6849111, https://www.researchgate.net/publication/271472780, [Retrieved:April, 2017].

[18] K. Zeng, L. Hou, H. Meng, Q. Zheng, and N. Lu, “Soft-Defined Heterogeneous Vehicular Network: Architecture and Challenges”, IEEE Network, vol. 30, pp.72-79, July/August 2016.

[19] K. Kai, W. Cong, and L. Tao, "Fog computing for vehicular Ad-hoc networks:paradigms, scenarios, and issues", The Journal of China Universities of Posts and

Telecommunications, www.sciencedirect.com/science/journal/10058885, http://jcupt.bupt.edu.cn, 23(2), pp.56–65, April 2016, [Retrieved: January, 2017].

[20] J. Chen, H. Zhou, N. Zhang, P. Yang, L. Gui, and X. Shen, ”Software defined Internet of vehicles: architecture, challenges and solutions”, Journal of Communications and Information Networks Vol. 1, Issue (1): pp.14-26, 2016, DOI: 10.11959/j.issn.2096-1081.2016.002.

[21] M. Patel et al., “Mobile-Edge Computing – Introductory Technical White Paper”, https://portal.etsi.org/Portals/0/TBpages/MEC/Docs/Mobile-edge_Computing_-Introductory_Technical_White_Paper_V1%2018-09-14.pdf/01.01.01_60/gs_MEC001v010101p.pdf [Retrieved: December, 2017].

[22] “Mobile edge computing (MEC); Framework and reference architecture,” ETSI, Sophia Antipolis, France, Mar. 2016. Available: http://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/01.01.01_60/gs_MEC003v010101p.pdf [Retrieved: January , 2018].

[23] T. Taleb,K. Samdanis, B. Mada, H. Flinck, S. Dutta, and Dario Sabella, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration", IEEE Comm. Surveys &Tutorials, Vol19, No.3, pp.1657-1681, 2017

[24] K. Zhang, Y. Mao, S. Leng,Y. He, and Y. Zhang, "Mobile-Edge Computing for Vehicular Networks- A Promising Network Paradigm with Predictive Off-Loading", IEEE vehicular technology magazine, pp.36-44, June 2017.

[25] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song, and M. Qiu, "A Scalable and Quick-Response Software Defined Vehicular Network Assisted by Mobile Edge Computing", IEEE Communications Magazine, pp.94-100, July 2017.