Social Internet of Vehicles: Architecture and enabling technologies

Article *in* Computers & Electrical Engineering · July 2018

DOI: 10.1016/j.compeleceng.2018.05.023

4 authors, including:

Razi Iqbal
University of Engineering and Technology
92 PUBLICATIONS **1,911** CITATIONS

SEE PROFILE

Contents lists available at ScienceDirect

Computers and Electrical Engineering

Journal homepage: www.elsevier.com/locate/compeleceng

Social Internet of Vehicles: Architecture and Enabling Technologies

Talal Ashraf Butt[a], Razi Iqbal[a], Sayed Chhattan Shah[b], Tariq Umar[c]

[a]*College of Computer Information Technology, American University in the Emirates, Dubai, United Arab Emirates*
[b]*Department of Information Communication Engineering, Hankuk University of Foreign Studies, Seoul, Korea.*
[c]*Department of Computer Science, COMSATS Institute of Information Technology, Wah Cantt. Pakistan.*

Abstract

The key goal of Internet of Things (IoT) has been the provision of value-added services based on the ubiquitously available smart devices that can offer diverse services by interacting with each other. However, the paradigm has evolved to its next phase, Social Internet of Things (SIoT), with the inception of an idea to empower these devices with consciousness. This cognizance enables these smart devices to socialize with each other based on shared context and mutual interests. The Social Internet of Vehicles (SIoV) applies SIoT concepts in the vehicular domain to revolutionize the existing ITS (Intelligent Transport System) by adding value to existing VANET (Vehicular Ad-hoc Network) technology. This paper presents a scalable SIoV architecture based on Restful web technology. Furthermore, this paper emphasizes the importance of web technology to meet the required interoperability to support the composition of numerous services. The paper also discusses the enabling technologies and protocols.

Keywords: VANET (Vehicular Ad-hoc Network); Internet of Vehicles; Social Internet of Vehicles; Internet of Things; Social Internet of Things; Web of Things; Intelligent transport systems

1. Introduction

VANETs (Vehicular Ad-hoc Networks) have seen impressive progress in recent decades due to escalation of communication technologies. Vehicular Networks have leveraged the benefits of various short range and long range wireless technologies for Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I) and Vehicle to Sensors (V2S) communication. Employment of Internet in Vehicular Networks has significantly enhanced the opportunities of developing applications for VANETs that seemed far-fetched in the past. Internet of Vehicles (IoV) [1] is hence emerged as an advancement to old-fashioned VANETs. IoV is conceptualized to solve several problems faced in traditional VANETs, such as, lack of coordination between disparate vehicles that are travelling at a distance from each other, scalability, ubiquity and information insufficiency, etc. In IoVs, each entity of the network can connect to the Internet. All time Internet connectivity brings the luxury of sharing information between different components of IoV network, e.g., Road Side Units (RSUs), vehicles, pedestrians, driver and passengers, etc. Besides information sharing, Internet connectivity provides the flexibility of widening the scale of the network.

Social Internet of Vehicle (SIoV) is the modern trend towards IoV [2]. In SIoV, entities socialize with each other by sharing information of common interests such as traffic information, weather conditions, road

situations, toll gates, vacant car parking slots and media sharing, etc. Socializing in SIoV is not limited to vehicles only, as the network can include drivers, passengers and infrastructure as well. The sharing of information in SIoV depends on several factors such as context, connection type, network structure, nature of application and environment. A SIoV system initiates at the manufacturing site of the vehicle. Once a vehicle is manufactured, it is equipped with sensors that can talk to the manufacturer for various operations such as maintenance and recovery. Subsequently, in SIoV, a vehicle maintains a social relationship list of other vehicles and talks to the owner through On-Board Unit (OBU) installed in the vehicle for sending and receiving information like navigation, etc. While on the road, a vehicle can communicate with other vehicles, infrastructures (RSUs) and pedestrians. Figure 1 illustrates the traditional SIoV model.

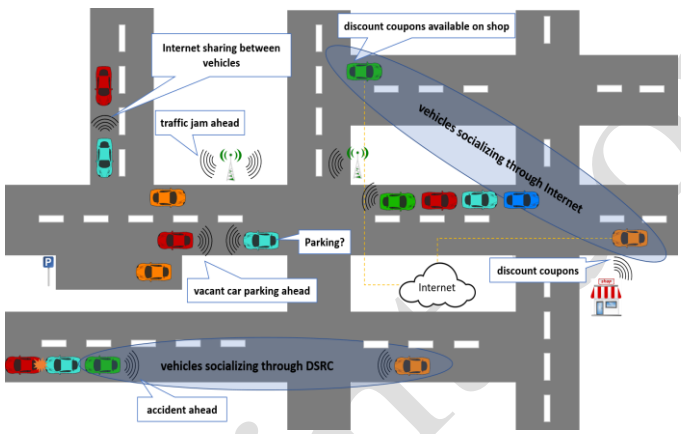


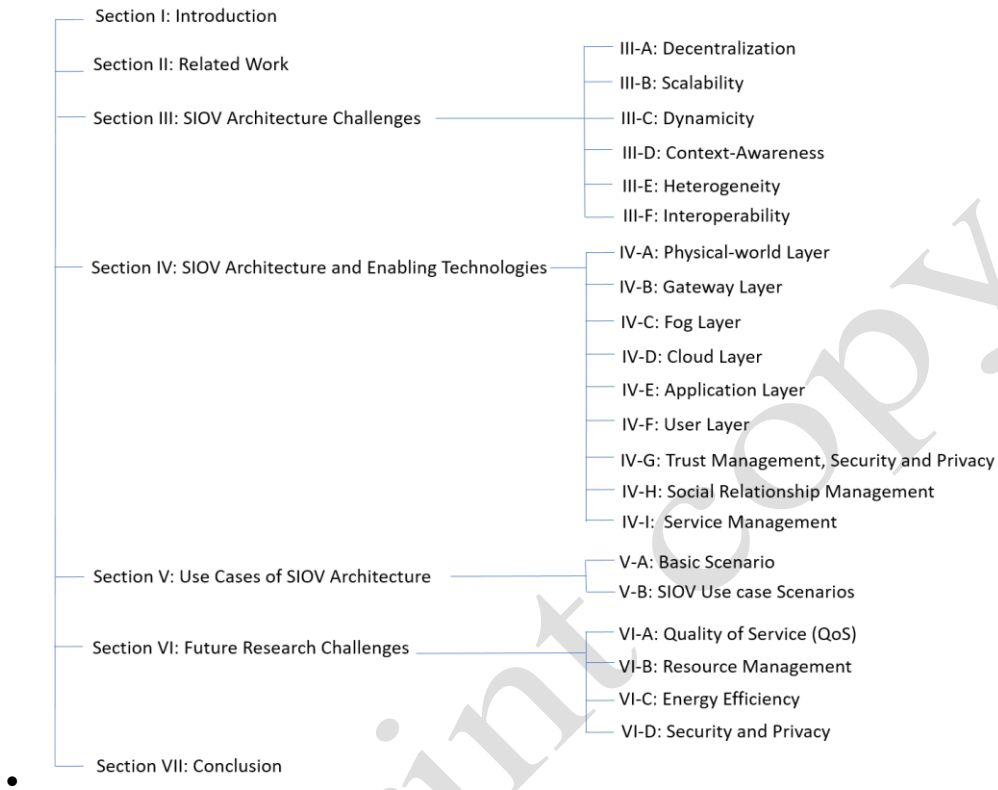
Figure 1: Traditional SIoV model

A key aspect of SIoV systems for socializing among its entities is centrality. It assists in providing a measure for finding entities that play a pivot role for a group of vehicles and facilitates efficient information dissemination by enabling the entities to act as a relay node. Clustering also plays a key role in socializing of entities in SIoV by categorizing vehicles based on parameters like interests of the vehicles, distance, speed and location. One of the major advantages of clustering is circumventing the broadcast storming problems along with assistance in increasing throughput and improving bit error rate.

One of the core objectives of SIoV is to focus on enabling the social relationships among various entities of the system. Mostly, these relationships are built on the context considering the mutual interests of the entities. For example, the transportation aspect of smart cities can be further enriched with the SIoV features by collecting real-time data from the connected vehicles based on their social relationships and taking smart decisions through intelligent analytics.

Nature of SIoV systems poses several challenges like dynamicity, interoperability, security, privacy, trust management, uncertainty, dependability, managing social relationship and heterogeneity, etc. Besides, these challenges, SIoV lacks a standard architecture. Recently, there has been efforts to develop a general architecture of IoV, however, a comprehensive understanding of SIoV architecture is still missing in the literature. This article is an effort towards proposing a general architecture of SIoV. The main contributions of this article are:

- Propose a scalable SIoV architecture based on Restful web technology to provide a foundation for developing SIoV applications.
- Emphasize on the importance of web technology to meet the required interoperability for supporting the composition of diverse services.
- Highlight the enabling technologies and protocols for SIoV systems.



- *Figure 2: Article's Organization*

The article's organization is presented in Figure 2. Section II reviews the related work conducted in the field of SIoV architecture. Section III discusses the challenges involved in designing the SIoV architecture. Section IV proposes a scalable SIoV architecture based on Restful web technology along with enabling technologies and discusses the service management for seamless integration of information provided by different entities of SIoV systems. Section V presents the use cases based on the proposed architecture to analyze its viability. Finally, Section VI discusses the future research challenges, and paper is concluded in Section VII.

2. Related Work

SIoV has the potential to enable new effective applications such as traffic safety, real-time control, and infotainment. Moreover, it allows businesses to benefit from the new paradigm by offering value-added services. The SIoV architecture requires addressing various issues such as heterogeneous devices and communication protocols in various domains, privacy, scalability and dependability.

The literature mostly addressed the general IoT requirements while defining the architecture [3][4] and few are focused on bringing web services based architecture to address the interoperability issue [5]. There are few efforts that focus on bringing the socializing factor to the IoT devices [6]. These efforts fail to address the specific requirements of SIoV. A considerable number of efforts tried to define an architecture for VANETs are available in the literature that mainly focus on V2V communication [7]. The SIoV paradigm requires an architecture that can address its specific requirements and should be general enough to cover its

various scenarios.

Some research efforts focus on defining architecture for IoV that is closely related to SIoV. However, the social relationship management aspect is remained unaddressed, which is the key for SIoV. In an attempt to cover various aspects of IoV, all of these efforts came up with relatively different IoV architectures that vary from three to seven layers. The authors in [8] proposed an IoV architecture with four layers: the base layer for embedded systems and sensors, the second layer allows the connectivity, the third layer consists of the core system to control and manage the system, and the fourth layer is the data centre cloud layer that hosts both public and private services. Similarly, the architecture defined in [9] has three layers that have almost the same functionalities, however, this one also defines load balancing at the cloud layer. In another effort [10], the authors rely mainly on mobile network enabled D2D (Device-to-Device) communication to define three layered architecture: first layer for end devices and D2D gateways, the second layer for Network management and third layer for applications. Another effort [11] proposes five-layered IoV architecture. The base perception layer consists of all sensors, vehicles and RSUs, and uses coordination layer to communicate with the cloud layer. The cloud layer offers the storage, processing and analysis of gathered data. The application layer consists of different applications that use cloud services through an interface, and business layer deals with the requirements of different business models. Recently, the authors in [12], [13] proposed a seven layered IoV architecture. They introduced a new base layer called User interaction that directly interacts with vehicle owners and manages notifications. The second layer combines the functionalities of the first two layers of previously discussed architectures.

There are few efforts in literature that cover the topic of enabling social relationships among devices. Nitti et.al [2] have focused on SIoV relationships, but explained only the ITS station architecture with a SIoV. The article lacks the detail of a holistic SIoV architecture beyond the realm of ITS station. Another effort [14] proposes a technique to securely maintain social relationships among vehicles, but provides no information regarding the architecture where the mechanism can be applicable. The authors in [15] emphasized the importance of establishing social relationships among vehicles and presented a schematic architecture of Vehicular Social Network (VSN). However, the detail about realizing the schematic architecture is not covered and layers are also not identified. Another article [16] provides a comprehensive survey of VSN and explains the importance of using and maintaining vehicular social relationships. However, the architecture details are not covered in the paper. Another effort [17] focuses on content dissemination in a VSN, but vaguely differentiate between a centralized and decentralized architecture without any concrete detail. Maglaras et al. [18] review the SIoV in the context of a smart city and focus on topics such as clustering of vehicles based on their social behaviour, security and privacy. The authors, however, have not covered architectural details of SIoV. A comprehensive research article [19] is the only effort in the literature that specifically presents a SIoV abstract architecture. The architecture consists of three layers: Physical, Cyber and Social layers. However, the contribution is more focused on building domain models and doesn't cover the architecture in detail.

The contribution of this paper is to fill the research gap by identifying the architectural challenges and proposes an extensible SIoV architecture to address these challenges. The paper also identifies and covers the details of the current technologies that can fit on each layer of the architecture for an effective SIoV system.

3. SIOV Architectural Challenges

SIoV poses several challenges when it comes to the design and development of a broader SIoV system. This section highlights the key challenges involved in designing a SIoV architecture along with crucial requirements in standardization, adaptability and infrastructure, etc.

3.1 Decentralization

Many SIoV applications have near real-time requirements that can be met with a decentralized architecture. The data gathered by highly mobile vehicles can become quickly invalid for many SIoV applications with a

change of context for any surrounding vehicle. The traditional centralized cloud architecture is not designed to address the SIoV latency and mobility requirements [20]. Furthermore, the data communication towards a cloud becomes a bottleneck for the efficiency during peak hours when the number of vehicles increases considerably. Therefore, SIoV needs a distributed and decentralized architecture to fulfil its latency, bandwidth and mobility requirements.

3.2 Scalability

The key advantage of SIoV over the traditional VANET is its flexibility in scale [16]. Internet connectivity of network entities enables distant peers to communicate with each other without being in the same fleet. Besides remote connections, scalability in SIoV provides information and resource rich environment that can offer on-demand services to various components of SIoV. For example, vehicles in SIoV can share road situations with each other on highways even if they are miles apart. Similarly, RSUs can provide vehicles entering the town with upcoming local events without having vehicles in clear line of sight. Scalability brings breadth to the SIoV without installing expensive infrastructures as parked vehicles can also act as RSUs if required. For example, if no nearby RSUs are available to provide passing vehicles information about vacant car parking, it can find an existing social relationship or create a new social relationship with a vehicle parked in the car park to share this information if it sees an unoccupied car parking. Such operations provide additional advantages to SIoV over traditional VANETs and IoV systems. A scalable SIoV undoubtedly augments the scope of the network, however, it also poses a challenge in designing the architecture of SIoV due to its dynamic expansion. Based on the operations of SIoV, it is difficult to predict the scale of the network as various components of the system enter and leave the network depending upon the context. The number of entities in a network can drastically change from dozens to hundreds in span of few seconds, hence, a robust, scalable and reliable architecture is required that can cope with expanding nature of SIoV systems.

3.3 Dynamicity

SIoV systems are highly dynamic in nature due to their mobile entities specially vehicles that have a tendency of changing their neighbouring nodes due to high speed. For example, a vehicle might have 2 RSUs and 5 vehicles as direct neighbours at one point, but this topology might change to 1 RSU and 20 vehicles within a span of few seconds [2]. In SIoV, message sharing between vehicles, RSUs, passengers and drivers, etc. occurs through V2V and V2I and is built on a dynamic relationship of the entities. This relationship can be a direct relationship or an indirect relationship through multi-hop communication. Ubiquitous nature of SIoV provides the flexibility of building long distance relationships between entities of the network that can enhance their trust in each other which ultimately strengthens the overall reputation of the network. However, the dynamicity in SIoVs affects the overall performance of the system in a dense environment. A network connected through a mesh topology having tens of RSUs and hundreds of vehicles, drivers and passengers might have to comprehensively reorganize itself if another vehicle enters this network (since this new entrant must communicate with other entities to socialize with them). For example, a vehicle entering in SIoV might need to develop a trust relationship with other entities of the network by socializing with at least a reasonable number of entities to form a good reputation. This reputation of the new vehicle must be updated in all the entities of SIoV to make it available for local and global access. This information dissemination entails substantial computing resources, timely processing and reliable communication which requires an efficient architecture for the system to avoid degradation in performance.

3.4 Context-Awareness

Information processing and sharing in SIoV is highly dependent on the context [9]. A crucial feature in SIoV is to enable vehicles, RSUs, drivers and passengers to be conscious about the situations around them,

especially those that are relevant to them and adapt their behaviour accordingly. For example, a RSU using the Internet, shares the information of a traffic jam on a road to vehicles that are miles away from it, based on the context, e.g., peak hours, days of the week (regular days, weekend, public holidays) and events, etc. This information might be different on different days and timings which makes information sharing in SIoV contingent to conditions, environment and situations. Context-awareness in SIoV is contemplated as salient facet as a slight miscalculation results in hazardous situations. For example, an inaccurate information of "no traffic jams" on a congested road provided to an ambulance carrying an emergency might result in unrecoverable loss. Such behaviour of the system might question its overall reliability, hence, a context-aware SIoV architecture is required that ensures coherence, consistency and reliability.

3.5 Heterogeneity

SIoV architecture needs to integrate heterogeneous vehicles and environmental sensors communicating using diverse communication technologies [16]. The vehicles in a safe environment vary from a small car to a truck with heavy loads. These heterogeneous vehicles have varied capabilities and sensors, and are interested in similar or drastically different applications. Moreover, these vehicles may support a range of different communication technologies that an architecture should consider. Furthermore, the environmental ITS-based sensors such as temperature, rain, motion, speed, parking, humidity, air quality, location and traffic light sensors are also needed to be included in the SIoV architecture, as these play a key role in many SIoV applications. These sensors when associated with infrastructure (RSU) enable applications like congestion detection, speed monitoring, vacant car parking detection, yellow line crossing and accident detection, etc. A future-proof SIoV architecture requires to be extensible by allowing easy adaptation of recent technologies.

3.6 Interoperability

SIoV consists of heterogeneous vehicles and sensors that use different communication protocols. Management of this heterogeneity and enabling the involved vehicles to communicate and understand each other is an important requirement for a SIoV architecture [1]. This requirement demands both syntactic and semantic interoperability to enable better collaboration among vehicles. Therefore, an interoperable architecture is required to seamlessly integrate these heterogeneous vehicles in to a system.

4. SIoV Architecture And Enabling Technologies

SIoV is emerging as an adherent to SIoT where vehicles can socialize with each other in order to share information, enhance capabilities and ensure safety on roads. SIoV is currently in its evolving phase that requires comprehensive research to develop standards, policies, rules, protocols and guidelines. Similarly, the literature lacks in defining a comprehensive architecture for SIoV that can set the foundation for development and deployment of SIoV applications and services. This section proposes the design of SIoV architecture while emphasizing on the latest trends in the technology and current available frameworks for similar systems like VANETs and IoVs.

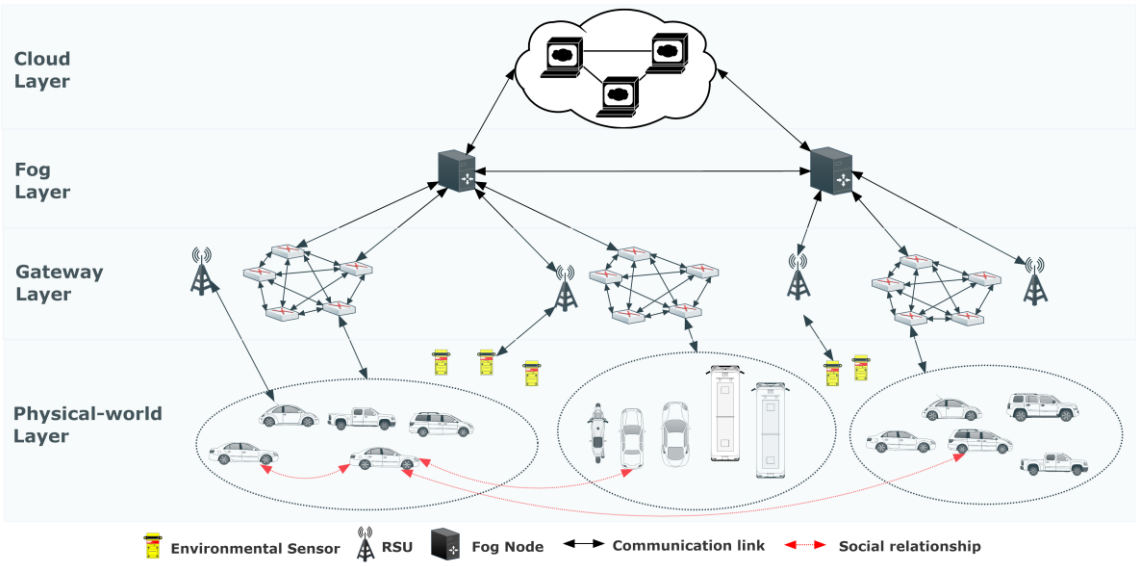


Figure 3: Holistic view of proposed SIoV architecture

Based on the gaps in the literature, this paper proposes an extensible and scalable SIoV architecture that addresses major potential challenges in SIoV. The existing solutions partially address few of the identified challenges, however, none of them confronts all the challenges presented in the previous section. Figure 3 shows a holistic view of the proposed architecture. Furthermore, the detail of the layered stack is shown in Figure 4, consists of Physical-world, Gateway, Fog, Cloud, and Application, User, Security Privacy and Trust Management and Relationship Management. In some cases, these layers can coexist on a single machine, e.g., when a vehicle offers functionality of a mobile fog unit to assist other vehicles [20]. Furthermore, the proposed SIoV architecture is extensible to integrate and work with the future trends such as Name Data Networking (NDN), Information Centric Networking (ICN), Software Defined Networking (SDN), and Network Function Virtualization (NFV). This section covers the details of these layers.

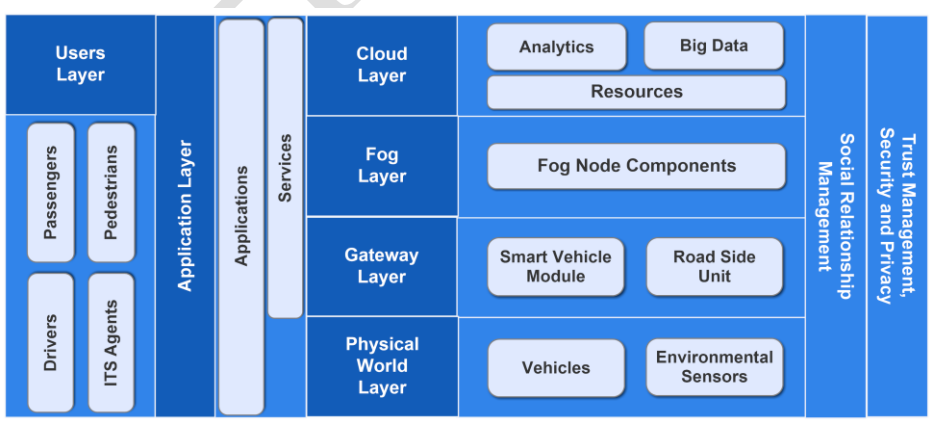


Figure 4: Proposed SIoV architecture

4.1 Physical-world Layer

The Physical-world layer of SIoV architecture deals with the physical objects like vehicles, environmental sensors, drivers, passengers and pedestrians, etc. The core functionality of this layer is to sense data through sensors installed in vehicles, and devices carried by drivers, passengers and pedestrians. It is a fundamental layer as it provides electrical, mechanical and intelligent interface to the overall architecture of SIoV. Besides being the most crucial layer of the architecture, it is also one of the most complex layers due to the variety of available technologies with broadly varying features. This section highlights the role of each entity by specifying their key functions performed in a typical SIoV environment.

***Vehicles*:** In SIoV architecture, a vehicle is considered the most crucial part of the system, as a single function performed by the vehicle can change the overall performance of SIoV system. A vehicle encompasses several sensors that enables On-Board Unit (OBU) of the vehicle to communicate with its different components, e.g., high beam sensors, parking sensors, blind spot sensors, driver seat sensors, airbag sensors, etc. Figure 5 illustrates some of the many sensors available in a vehicle.

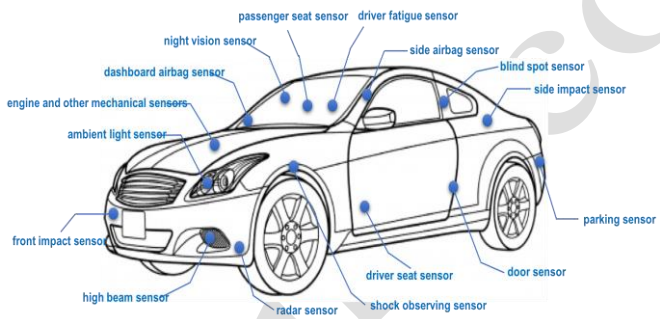


Figure 5: Potential sensors available in a Vehicle

An OBU in modern vehicles provides them additional functionality of intra-vehicle communication. An intra-vehicle communication is an interaction between OBU and vehicle sensors or OBU and drivers' or passengers' handheld devices. In order to maintain an efficient interaction between vehicle and its components, this article proposes a vehicle architecture based on review of literature. Figure 6 presents the proposed architecture of a vehicle for SIoV systems.

A Vehicle architecture comprises of four layers: Physical layer, Intra-Vehicle Communication layer, Processing layer and Application layer. The physical layer of vehicle architecture consists of various sensors like high beam sensor, rain sensor, parking sensor, airbag sensor, engine sensor and accelerometer sensor, etc. These sensors are required to sense the respective values that are then transmitted to OBU for processing and ultimately taking the appropriate actions. Modern vehicles are equipped with hundreds of such sensors. Different car manufacturers use different sensors to ensure high performance and efficient drive. As per the standard of Wireless Access in Vehicular Environment (WAVE), OBUs can communicate with each other over a fixed radio channel called a Control Channel (CCH). In order to standardize the communication between sensors and OBU, several protocols have been implemented at the physical layer of the vehicle architecture. Some of those protocols are: Physical Medium Dependent (PMD) (used to interface with the wireless medium and utilizes the Orthogonal Frequency Division Multiplexing (OFDM) for modulation, Physical Layer Convergence Procedure (PLCP) outlines the mapping between MAC frame and OFDM unit, Physical Layer Management Entity (PLME) assists in setting up the connection and managing WAVE compliant devices.

Intra-Vehicle communication layer is responsible for communication between the sensors, actuators

and radio with the central unit OBU. This layer regulates the communication technologies for intra-vehicle communication. Popular technologies used at this level are Bluetooth, irDa, Wi-Fi and ZigBee, etc. Several protocols work at this layer depending upon the technologies used for intra-vehicle communication. Some of the protocols used at this layer are RFCOMM (Radio Frequency Communication), LMP (Link Manager Protocol), L2CAP (Logical link control and adaptation protocol) and SDP (Service discovery protocol) etc. TCP/IP (Transmission Control Protocol/Internet Protocol) is widely used for Wi-Fi along with protocols for irDa that are IrLAP (Infrared Link Access Protocol), IrLMP (Infrared Link Management Protocol) and Tiny TP (Tiny Transport Protocol).

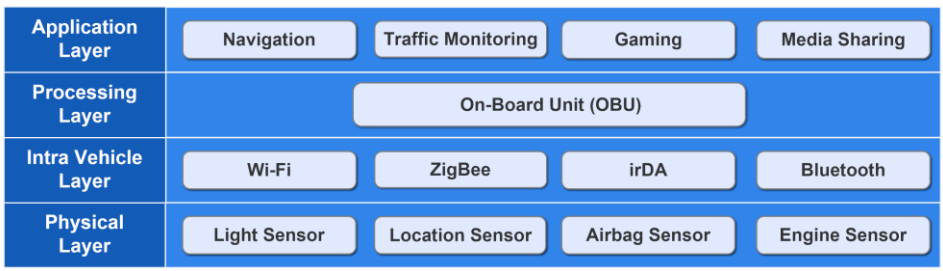


Figure 6: Vehicle Layered Architecture for SIoV Systems

Processing Layer is where all the computation and processing takes place. A special unit of the vehicle, OBU is in-charge of processing the information received from intra-vehicle layer. An OBU is logically comprised of a wireless communication technology like DSRC, RFID or ZigBee, etc., a GPS system, a CPU for application processing and interfaces to in-vehicle sensors. OBUs transfer information to other OBUs and at times responsible for collecting data to support public applications. OBUs besides processing and transferring of data, store information as well to provide quick response to local data requests, e.g., vehicle registration number, past travel history, owners' data and vehicle manufacturers' details, etc. A Controller Area Network (CAN) operates at this layer to network intelligent devices and helps in linking the ECUs (Electronic Controller Units) of different devices in order to exchange real-time information. A CAN protocol helps in reducing the complex wiring, meeting time constraints, and error free transmission. Besides CAN protocol, several protocols are utilized at this layer: CAN protocol is used for communication between OBU and vehicle sensors as mentioned earlier, RTP (Real-Time Transport Protocol) enables the vehicle to deliver audio or video over the IP network to socialize with peer vehicles and other components of the network. Besides RTP, this layer also exploits RTSP (Real-Time Streaming Protocol) which is used to control streaming media servers. In SIoV, each entity can serve as a web-server, hence this protocol is used by OBU in association with Gateway Layer to provide web-server facilities to various in-vehicle components and other peer-vehicles that are not capable of connecting to the Internet. CIP (Common Industrial Protocol) is another set of services and messages used in this layer for collection of manufacturing automation application such as safety, control and motion. This protocol enables users to integrate these manufacturing applications with the Internet. Finally, IPSec is used for secure intra-vehicle and inter-vehicle communication. It supports data-integrity, authentication, replay protection and data confidentiality.

Environmental Sensors: An essential function of the physical-world layer of SIoV architecture is sensing that assists in gathering data from various environmental sensors. The information collected from these sensors ultimately helps in taking required actions. Several environmental sensors are used to gather distinct information for processing, computing and communicating. Some of the widely used sensors are, temperature, rain, motion, speed, parking, yellow line, microphone, humidity, air quality, location and traffic

light sensors, etc. These sensors when associated with infrastructure (RSU) enable applications such as congestion detection, speed monitoring, vacant car parking detection, yellow line crossing and accident detection. Figure 7 illustrates a few of the environmental sensors in various SIoV scenarios

Environmental sensors operate through several protocols depending upon their nature. For example, temperature and motion sensors use DCON protocol that uses a master line to send commands to the client device by referring them through a unique address. Besides using DCON protocol, these sensors also utilize ModBus RTU (Remote Terminal Unit) for serial communication that allows techniques like CRC (Cyclic Redundancy Check) for error detection and correction. Similarly, location, speed and sonar sensors use the NMEA 0183 protocol developed and maintained by the National Marine Electronics Association. It uses a serial communication protocol that defines the transmission of data in the form of a sentence from a sender to multiple receivers. Temperature and Humidity sensors are widely used on roads to display temperature and humidity levels; these sensors use 1-wire and I2C protocols that are suitable for low data rates. Speed monitoring of vehicles on roads is done by measuring the travel time of the vehicle between two fixed points using methods like Doppler radar and LIDAR (Light Imaging, Detection, And Ranging). These techniques use different protocols for communicating with other processing and computing units, e.g., the Doppler radar technique uses COP (Common Output Protocol) for sending speed information to devices like in-vehicle processing and display units that can further process the information to take appropriate actions.

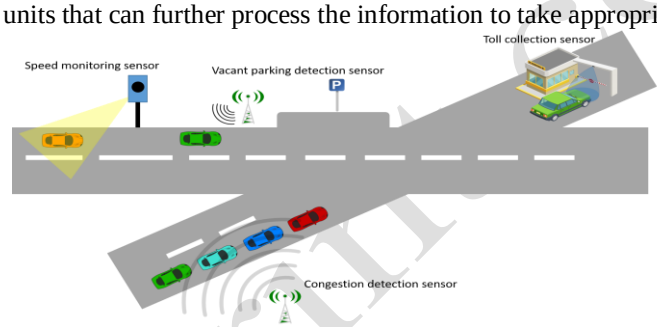


Figure 7: Environmental Sensors in SIoV systems

As mentioned earlier, the core functionality of environmental sensors is to gather information which is later processed, computed and communicated by the associated units like processors and network devices to take the required action. Table 1 illustrates the functioning and types of most common protocols at Physical World Layer of the proposed SIoV architecture.

4.2 Gateway Layer

The gateway layer is responsible to provide a portal to the physical-world layer towards the cloud-based infrastructure. This layer gathers the physical-world data and passes it to the fog layer. All the data are communicated to the smart vehicle module by intra-vehicle network using Bluetooth, BLE or Wi-Fi. Smart vehicles have special modules with necessary communication protocol stacks that allow them to directly talk to the fog layer. On the other hand, traditional vehicles and environmental sensors that don't have direct means of sending data to fog layer use RSU or nearby smart vehicle module to do so. Figure 8 presents the architecture of the gateway layer. The architecture consists of three layers: Integration, Middleware and Application layers. The integration layer enables the smart vehicle module or RSU to communicate with other traditional vehicles and environmental sensors. This communication mainly relies on common standard technologies including Low Power Wide-area Network (LPWAN), DSRC (Dedicated Short-Range Communication) and 6LoWPAN. The functionality of middleware and application layers differ for a smart car module and RSU.

Table 1: Protocols at Physical World Layer

Protocols	Communication	Type	Description
PMD	Intra-Vehicle	Interfacing protocol	Assists in interfacing with wireless medium within a vehicle
PLCP	Intra-Vehicle	Mapping protocol	Helps in mapping between MAC frame and ODM unit for various physical layer units' communication
PLME	Intra-Vehicle	Connection protocol	Supports in establishing a connection of WAVE compliance devices for intra-vehicle communication
CAN	Intra-Vehicle / V2S	Communication protocol	Contributes in communication between Vehicle sensors and On-Board Unit
RTP	Intra-Vehicle	Communication protocol	Assists in delivery of Audio Video content over IP within vehicle
RTSP	Intra-Vehicle	Communication / Streaming protocol	Helps in controlling streaming media servers for various media within vehicle
CIP	Intra-Vehicle / V2S	Control and Information Protocol	Supports collection of information from various vehicle sensors for manufacturing automation
I2C	V2S	Intra-Board Communication	Contributes in interfacing various vehicle modules to On-Board Unit

A smart vehicle module can process its collected data and making that data available to other smart vehicles in the form of web services. The middleware layer is focused on pre-processing the data and managing the web services. In SIoV, it is essential to efficiently create and manage social relationships. The middleware layer performs this function. These social relationships are dynamically updated according to the context of a vehicle, e.g., if a vehicle takes an exit on a highway then the relationships with the previous neighbours will not be of great interest anymore. A Service API enables different applications at the application layer to consume the managed services. A smart car can support all the V2V applications mentioned in the physical-world layer while offering the required interoperability enabled by web services. Inter-Vehicle Communication deals with the interaction of vehicles with each other. Besides V2V communication, it also deals with V2I communication. Several communication technologies are used at this layer to enable socialization of a vehicle with other entities of SIoV, e.g., DSRC, RFID (Radio Frequency Identification) and ZigBee, etc. DSRC is specifically designed to work with vehicular systems as it meets the low latency requirement for road safety messaging and control. DSRC protocol architecture is a combination of IEEE 1609.0, IEEE 1609.1, IEEE 1609.2, IEEE 1609.3, IEEE 1609.4, and IEEE802.11p.

Figure 8: Architecture of Gateway Layer

Table 2. Protocols at Gateway Layer

Protocols	Communication	Protocol Type	Description
MAC	V2V / V2I	QoS	Assists in avoiding and resolve packet collisions in multi-lane environments
LLC	V2V / V2I	Connection	Helps in establishing connection-less and connection-oriented services between Vehicles and Infrastructures etc.
WME	V2V / V2I	Processing	Supports in performing management and processing of frame queuing, prioritizing the channels and handling of safety messages.
WSMP	V2V / V2I	Transmission	Contributes in exchange of safety information along with transmitting the packets at a low power or data rate
TCP/IP	V2V / V2I/V2P	Communication	Assists in communicating with local network or the Internet

The protocols used by DSRC are MAC protocol that assists in avoiding and resolve packet collisions in multi-lane environments, LLC protocol that helps in establishing connectionless and connection-oriented services, WME (WAVE Management Entity) performs management of the frame queuing, prioritizing the channels and handling of safety messages, WSMP (WAVE Short Messages Protocol) that is used to exchange safety information between V2V and V2I along with providing special services like transmitting the packets at a low power or data rate. Finally, this communication employs TCP/IP to utilize local network services through either the Smart Vehicle Module or RSU (Gateway Layer), which enables vehicles and RSUs to create their own fragmented network if large network is not available. Furthermore, the fog agent at this layer enables a smart vehicle to send updates to the fog layer using the available cellular or other networks. Table 2 illustrates the functionality and description of protocols used at this layer in SIoV architecture. The role of middleware layer is similar for RSUs, but these deal with various traditional vehicles and environmental sensors. Each RSU collects and forwards data on behalf of several vehicles and environmental sensors in its vicinity. It can also perform some processing or aggregation of the collected data and expose that data in the form of web services. Moreover, it dynamically manages the social relationships of multiple traditional vehicles.

Several applications are enabled at RSU such as safety, efficiency and infotainment applications. In contrast to V2V real-time safety applications, the role of gateway based applications is to serve soft real-time goals such as local road and weather conditions, smart traffic signs, and traffic congestion state. The traffic efficiency applications rely on the sharing the pre-processed gathered data from several vehicles among the vehicles in the vicinity. Furthermore, the applications can use this data to manage speed of vehicles.

4.3 Fog Layer

The fog layer integrates the Fog computing paradigm [21] in SIoV architecture. The concept of Fog Computing was coined by CISCO in 2012. Its goal is to bring the extension of the cloud computing paradigm at the edge of a network closer to the end-user devices. This paradigm is tightly linked with the existing cloud technology. Furthermore, the fog has multi-tier architecture and therefore offers more flexibility compared to Mobile Edge Computing (MEC) and Cloudlet technologies. The fog layer consists of many fog nodes that can be edge routers, smartphones and various other computing systems. This layer plays a vital role in ensuring the required scalability with the provision of fog-based decentralized architecture in a SIoV where millions of smart vehicles generating a vast amount of data. Figure 9 presents the architecture of the Fog layer based on the OpenFog standard guidelines [22]. Furthermore, Table 3 illustrates the enabling technologies of fog layer. It

TABLE 3. Protocols at Fog Layer

Category	Protocols	Description
Communication	Wi-Fi	Low-range, High bandwidth
	WiMax	Wide range, High bandwidth and energy cost
	DSRC	Specialized vehicular short distance communication
	LPWAN: SigFox, LoRa	Energy efficient, high bandwidth
	Cellular (LTE-A)	Wide range, High bandwidth and
IP	6LoWPAN	Compressed IPv6 for constrained devices
Transport	TCP	Single stream transmission
	SCTP	Multi-tenant streaming
	UDP	Individual datagram communication

uses wide-range of communication protocols including WiFi, WiMax, DSRC, WPAN, and cellular technologies to integrate plethora of devices with diverse capabilities. It also employs 6LoWPAN technology to corporate with constrained devices. Moreover, it uses range of Transport protocols such as TCP, User Datagram Protocol (UDP), and Stream Control Transmission Protocol (SCTP), and Application protocols including Constrained Application Protocol (COAP), Message Queuing Telemetry Transport (MQTT), and Extensible Messaging and Presence Protocol (XMPP) to meet the requirements of various applications.

The fog layer focuses on storing a substantial amount of data close to the end-user rather than sending it across the Internet towards the centralized data centres. This reduces the communication time required for the data to route through the Internet. The network management, including control and configuration is also distributed among the fog nodes. These fog nodes can be deployed at a fixed location or can be placed on a vehicle. The fog nodes are distributed to manage different areas of SIoV. The nodes could be deployed in a multiple tier to cover a small neighbourhood to a region based on the coverage requirements of SIoV applications. Furthermore, these nodes can form a collaborative cluster based on their context. The dynamic clustering can enable the computational load balancing to utilize more resources simultaneously.

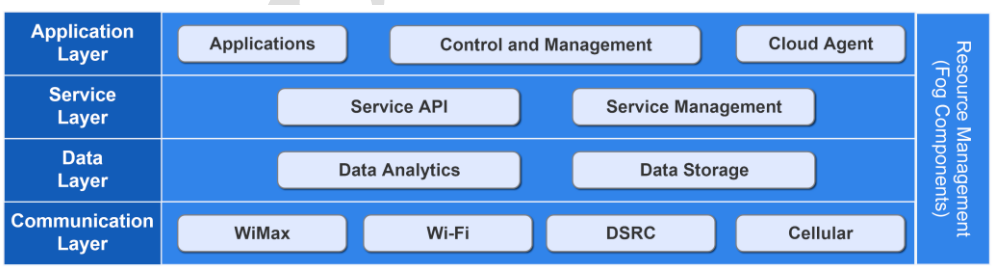


Figure 9: Architecture of Fog Layer

Each node at fog layer receives raw or pre-processed data from the Gateway layer using available communication protocols. At data layer, these data are stored in the node and further analyzed to offer real-time data analysis for delay-sensitive applications. This layer also temporarily stores and manages the records of social relationships between different vehicles within the area of the Fog node. The Service layer creates and manages services for the interoperable consumption of the data through its Service API. The application layer hosts different applications and control & management interface. A cloud agent also runs at this layer to maintain a connection with cloud. The cloud agent interacts and cooperates with cloud to get assistance to deal with situations where the fog node requires more capabilities in terms of computing and storage. The

resource management layer fulfils the extra demand of resources by either locally dedicating more resources or collaborating with other fog nodes for computation load balancing.

Fog layer supports all applications that require real-time Quality of Service (QoS) with low latency, location awareness, mobility, and low energy consumption. In SIoV, several applications such as intelligent parking, Platooning, Content delivery, and local Traffic information need fog layer support to run.

4.4 Cloud Layer

The cloud layer takes advantage of cloud technologies to provide a centralized backend that is empowered by highly capable servers and storage. It provides resources to perform complex computations, store massive amounts of data and a place to make system-wide decisions. The cloud layer dynamically schedules and manages its resources in response to the devised policies and current trends of the system. The architecture of the cloud layer is like the fog layer with the difference of scale of permanently stored data, as shown in Figure 10.

The cloud layer is the centralized hub that stores all the data shared by vehicles and their social relationships, and environmental sensors. It manages resources by dynamically allocating new servers or adding new storage capabilities using Sensing, Infrastructure, Platform, Network, and Storage as a Service according to the demand of a system. The Infrastructure is virtualized using OpenStack, OpenNebula and CloudStack platforms. There are commercial Storage and Platform virtualization solutions available including Amazon S3, Amazon EC2 and Google AppEngine. OpenIoT is employed for the provision of sensors as services to fulfil the requirements of diverse applications. Furthermore, there are many research efforts are utilizing Software Defined Networking (SDN), Network Function Virtualization (NFV) to offer on demand network services.

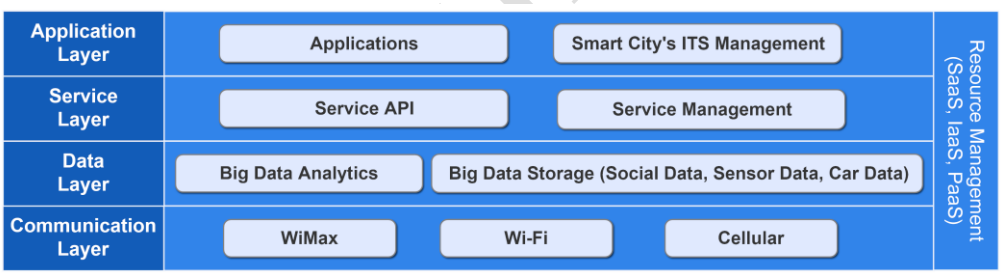


Figure 10: Architecture of cloud layer

The cloud layer offers resources as Web Services by either offering Service Oriented Architecture (SOA), or Representational State Transfer (REST) Architecture. The SOA employs Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), and Universal Description, Discovery, and Integration (UDDI). Whereas, the REST utilizes COAP or Hyper Text Terminal Protocol (HTTP). It utilizes Big data technologies to deal with the storage and management of these massive data [23] Furthermore, it uses Big data analytics to extract useful information from the data. Table 4 contains the list of different platforms and solutions currently available for distributed storage, batch processing and stream processing of big data. Apache Hadoop HDFS and Apache HBase are the open-source solutions for distributed storage. The cloud processes batch files using Apache MapReduce and Apache Spark. However, the stream processing is done by solutions such as Apache Storm, and Apache Spark Streaming.

The social relationships data enrich the analytics to find certain patterns that are otherwise not possible. For example, two vehicles who have social relationships between them will affect the traffic planning application even when they are travelling on disjointed roads for some time. The service layer creates and

Table 4: Protocols and Platforms at Cloud layer

Technology	Category	Protocols and Platforms
Virtualization	Sensing as a Service	OpenIoT
	Infrastructure as a Service	OpenStack, OpenNebula, CloudStack
	Storage as a Service	Amazon S3
	Network as a Service	Software Defined Networking (SDN), Network Function Virtualization (NFV)
	Platform as a Service	Amazon EC2, Google AppEngine
Web Services	Service Oriented Architecture (SOA)	Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), Universal Description, Discovery, and Integration (UDDI)
	Representational State Transfer (REST)	COAP, HTTP
Data Analytics	Distributed Storage	Apache Hadoop HDFS, Apache HBase
	Batch Processing	Apache MapReduce, Apache Spark
	Stream Processing	Apache Storm, Apache Spark Streaming

manages value-added services and exposes them using a Service API. The cloud layer serves all applications that require high computation or need results based on some past data. The Smart City's ITS management applications such as Traffic control, Traffic predictions and requirements are inherently based on cloud layer.

4.5 Application Layer

This layer is responsible for providing application specific services to the user. Several applications can be defined in this layer such as navigation, social, infotainment, safety and utility. In SIoV systems, the application layer is expected to play a vital role at all the layers of the architecture based on nature and context of the

applications. Based on the services provided by the application layer, Table 5 illustrates the details of different application protocols that can be used in SIoV. Several protocols are utilized at this layer based on the type of application, e.g., HTTP is used as a defacto for RESTful web services, CoAP is used for communication between low power and low memory devices, MQTT used for its simplicity and low requirements of CPU and memory, Advanced Message Queuing Protocol (AMQP) assists in storing data at the time of network disruption, Web-Socket protocol that delivers two-way communication between clients and a remote server on single CP channel, Extensible Messaging and Presence Protocol (XMPP) for chatting and message exchange applications, Data Distribution Service (DDS) for M2M (Machine to Machine) communications and Secure MQTT (SMQTT) that uses secure communication based on encryption. The websockets are used to reduce the overall communication required over the Internet. Furthermore, websockets employ WAMP (Websocket Application Messaging Protocol) to enable publish/subscribe messaging system. The proposed SIoV architecture utilizes the services of the application layer at all the layers of the architecture which makes the scope of this layer system-wide; whether its intra-vehicle infotainment apps at physical-world layer or toll collection apps at gateway layer, congestion monitoring apps at fog layer or data analytics app at cloud layer, application layer provides its required services.

4.6 User Layer

SIoV systems distinguish themselves from VANETs and IoV in socializing aspect. Vehicles and infrastructures can socialize with each other by sharing information of common interest. Table 6 provides the

Table 5: Protocols at Application Layer

Protocols	Description	Transport	Security
HTTP	A defacto standard protocol to enable RESTFUL Services (Representational State Transfer).	TCP	HTTPS
CoAP (Constrained Application Protocol)	A synchronous request/response application-layer protocol that is used as an alternative of HTTP by constrained devices.	UDP	DTLS
MQTT (Message Queue Telemetry Transport)	A broker based asynchronous publish/subscribe application protocol.	TCP	TLS/SSL

list of users and applications of SIoV and highlights the requirements of the applications. Besides vehicle and infrastructure other entities like drivers, passengers and pedestrians play a vital role in SIoV architecture as they are key contributors to the system when it comes to socialization. These entities socialize with each other and vehicle and infrastructure through handheld devices like smart phones, tablets and wearables such as smart watches. Drivers can socialize with vehicles by sharing a destination information, personal details, financial details (for toll payment), fatigue level, health conditions, contact details and music interests, etc. Similarly, a vehicle can socialize with its driver by sharing information such as percentage of fuel left, cabin temperature, car condition, nearby restaurants and gas stations. Besides communicating with each other, vehicles and drivers can communicate with passengers in the car. For example, drivers and passengers can socialize with each other by sharing contact details, media, the Internet, fatigue level, network (free minutes and messages) and sensor data and vice versa. Similarly, passengers can communicate with a vehicle by sharing information such as personal details, health conditions and music interests. Passengers can communicate with each other by exchanging information like, chats, video sharing, gaming, etc.

Socializing becomes purposeful when utilized in public transports where several passengers are higher than private transports. Besides the number of passengers, public transport might have more information to share as it is governed by law enforcing agencies. The passengers in a public transport like a bus might have different options of connecting to the Internet, e.g., direct internet connection through the mobile service providers or internet service provided by the bus. A passenger connected to the bus network might be entitled to receive notifications like bus schedules, free tickets, upcoming town events, local town news, changes in traffic laws, traffic and weather conditions, bus fares, etc. Passengers would still be able to socialize through social networking websites available on the Internet as bus provides a hotspot for internet connectivity as illustrated in Figure 11.

Pedestrians socialize with different entities of the SIoV system through handheld devices like drivers and passengers. They share information like location, personal details, social profiles, food interests and preferred movie genres, etc. Based on the information shared by the pedestrians, SIoV entities provide relevant information for the pedestrians. Another key component of User Layer is ITS agent that involves users from law enforcing agencies and third-party developers. The significant role of this agent is to provide applications and services to different entities of the system. It encompasses the monitoring and surveillance of roads through control centres of law enforcing agencies, along with agents of third parties like restaurants, service providers and merchants, etc. that provide offers to the users of the roads.

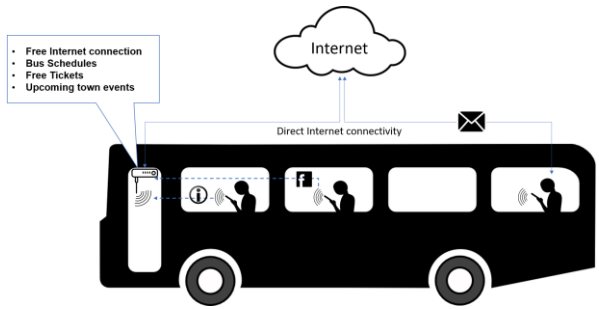


Figure 11: Passengers socializing in public transport

4.7 Trust Management, Security and Privacy

SIoV systems are highly connected systems that require extreme security and privacy measures to protect the data of not only the entities involved in the system but also applications and services utilized by them. This layer deals with Security, Privacy and Trust Management of SIoV entities and hence plays a significant role at all the layers of the SIoV architecture. A minor breach at this layer can affect the whole architecture and might lead to life-threatening incidents.

The security is a crucial aspect of SIoV, as the compromise of any entity of the system can lead to catastrophic situations. Majority of VANET and IoT attacks are valid for SIoV as they share the same fundamental principles of connected objects. Literature provides several security attacks on VANETs, IoVs and SIoVs, such as impersonation, DoS (Denial of Service), Eavesdropping, False message injection, Masquerading and Sybil and malware attacks, etc. Broad connectivity of SIoV entities in wide geographic locations along with the heterogeneous nature regarding architecture, type, ownership, region, manufacturer and users makes security quite challenging.

Socializing requires sharing of information that might include personal details. In SIoV systems, entities are dynamically connected to share information of common interests that can be processed to provide useful facilities to peer entities of the network [2]. Besides sharing personal information of drivers and passengers, vehicles might share information like, location, speed, source, destination and sensor information to other entities of the system. The confidential information that is used to offer safety applications can be misused by third parties to infer the behaviour of the entities based on their location history, duration of a visit and time spent per visit, etc. In SIoV systems, during wireless communication, privacy is quite challenging due to the scalability of network, visibility of objects, high changes of infidelity, the anonymity of the entities and enormity of data. Privacy protection for SIoVs should be based on efficient technologies for social and vehicular networks as SIoV is a combination of both social and vehicular systems.

Trust management in SIoV is of utmost importance because of dynamic topologies, uncertainty, subjectivity, intransitivity, context dependence and non-cooperativeness of entities in certain situations. Trust in SIoV systems depends upon the type of relationship and number of interactions between the entities. In SIoV, the system expects that entities establish trust through several interactions and that it can be testified to affect the upcoming communications or recommendations for that entity that builds its reputation. Trust in general depends upon reputation based systems that are heavily dependent rely on certainty, rationality, predictability, and recurrence of behaviours.

The significance of security, privacy and trust management at different layers of SIoV's architecture is essential for its reliability. At the physical-world layer, vehicles and environmental sensors need to be secured because these provide critical information required by different entities of the SIoV architecture. In order to ensure proper security of the sensor data and integrity of a vehicle, confidential data and trustworthiness of the information provided by the vehicle, strict security, privacy and trust rules should be applied at this layer.

Table 6: User Layer Application requirements

Users at Layer	Applications	Requirements			
		Latency	Bandwidth	Data	
				Source	Time
Physical-world layer: Driver, Passengers	Vehicle Safety	Real-time	Low	In-car sensors	Milliseconds to seconds
	Driver Health Monitoring				
	In-Car Infotainment		High		
	Multi-passenger Gaming				
Gateway layer: Drivers, Passengers	Roadworks warning	Real-time	Low	Other neighbour vehicles	Seconds to minutes
	Traffic Congestions				
	Live Traffic Conditions				
	Platooning				
	Point-of-interest Notifications	Low	Medium		
	Vacant Parking Information				
	Neighbour Driver collaboration (Verse)				
Fog layer: Drivers, Smart cities users	Traffic Light Management	Low	Low	Route-area	Up to few hours
	Vehicle tracking				
	Driver chat (RoadSpeak)				
	Road Safety Messages (SBone, ICNow)				
	Accident Warnings within an area				
	Emergency Vehicle Warning				
	Commercial Advertisement				
	Toll Collection				
	Alternate Routes (Drive and Share)	Medium	Low		
Cloud layer: Smart cities users	Traffic management	Medium to high	Low	Whole network	Few days to months
	Toll Collection updates				
	Driver Behaviour (SocialDrive)				
	Maps Update (Waze, MobiliNet)				
	Highway Information (Moovit, GasBuddy)				
	Weather Information				
	Multimedia File Sharing				

At the gateway layer, smart vehicle modules and RSUs are responsible for routing and forwarding of information received from the physical-world layer require uncompromising security, privacy and trust management. Failure to abide by rules at this layer might not only affect that segment of the network, but also a whole network by broadcasting false information about entities of the system. The security, privacy and trust at the fog layer play a substantial role as it emphasizes on storing extensive amount of data close to the end-user rather than sending it across the Internet towards the centralized data centres [24]. Due to substantial amounts of data stored in fog nodes, security becomes paramount as a small breach at this point may result in compromise of enormous information that would ultimately jeopardize the entire network. The cloud layer

applications and services require unswerving policies of security, privacy and trust [25]. A slight miscalculation in trust of SIoV entities in this layer would result in uncertainty in the behaviour of other entities towards each other. Similarly, breach of personal data in this layer would affect the overall reputation of the system. At the application layer, security, privacy and trust are extremely vital as this layer is the closest to the end user, e.g., vehicle, driver, passenger and pedestrian, etc. Enhanced security, absolute privacy and unquestionable trust are highly desirable at this layer. An attack on the personal details of the user might not only result in compromise of confidential data, but also affect the general reputation of the system.

4.8 Social Relationship Management

The idea of SIoV is based on allowing vehicles to create social relationships with other vehicles based on mutual interests. These vehicles share services between each other, delegate tasks and to create new services by collaborating to achieve tasks that are beyond their individual capabilities. These relationships are not only dependent on the location, as other factors such as travelling to the same destination play a key role in the creation of new relationships. Atzori et al. [6] have identified these relationships in four categories. Parental Object Relationships (POR) depend on homogeneity based on same manufacturer. A vehicle can contact other vehicles in its POR social list to resolve an issue that they may have solved. Co-location Object Relationships (COR) are concerned with the same location. COR based social relationships can assist a vehicle to get an instant update from other vehicles in the same vicinity. Co-Work Object Relationships (C-WOR) depend on being part of one application or solving the same problem. These relationships are maintained by vehicles, regardless of their locations based on mutual interests. Ownership Object Relationships (OOR) are established based on the object owner's social relationships. We are taking a step further to categorize these as short-term and long-term relationships based on the validity period of a relationship. Short-term relationships are saved, used, and synchronized to fog from each smart vehicle module. After a journey, a vehicle either saves a short-term relationship as a long-term relationship based on certain criteria chosen by its owner or deletes its record. POR and OOR relationships are considered long-term relationships by default. Long-term relationships are maintained and eventually synchronized to cloud as a backup. On the other hand, each fog node temporarily maintains a record of current vehicles and their social relationships record until the vehicle is in its vicinity.

4.9 Service Management

The SIoV paradigm requires network navigability to discover and use services based on trustworthiness to serve requirements of different applications. However, the paradigm consists of heterogeneous vehicles, sensors, and communication protocols. This demands a solution that can sufficiently deal with the inherent challenge of interoperability between different entities. We propose to use Restful web services in order to seamlessly integrate a diverse range of vehicles and information sources in this paradigm. In order to enable semantic interoperability, we advocate the utilization of semantic web based ontologies to describe the web services. In our proposed architecture, the service manager at different layers of SIoV leverage web standards for Restful service creation, management and sharing to serve a variety of applications while managing the social relationships. This transition evolves the SIoV to Social Web of Vehicles (SWoV). The SWoV paradigm enables the smart vehicle modules to run a small HTTP or CoAP web server with that offers and interoperable web services, based on the available sensor readings, ready to be shared with other smart modules in their social circles. After sharing, other modules can directly access these services using RESTful APIs. For example, a vehicle can consume services of other social contacts travelling within the context to get road works ahead. These vehicles can collaborate to compose complex services by creating mashups of available web services. Moreover, these smart modules will keep the fog node updated to assist their contacts to avoid a traffic jam who are travelling on different routes or miles away from them. At cloud layer, the services are offered by the virtual counterparts of physical vehicles for more complex value-added services.

5. Use Cases of SIoV Architecture

This section covers the description of few vehicular applications use case scenarios to analyze the proposed SIoV architecture works at an acceptable level. The use cases emphasize the benefit of using applications at different layers of the architecture. The first use case scenario focuses on a key application of a smart vehicle module and meets the required latency for the real-time application. Whereas, the second use case scenario relies on the distributed fog layer to provide the necessary aid to meet the requirements of the application. Lastly, our third use case scenario demands resources and information that are fulfilled by cloud. The use cases are based on a basic scenario that will be improved by different SIoV applications.

5.1 Basic Scenario

Prof. Adel, a professor at American University in the Emirates, drives around 40 km daily to and from his home to his workplace. He is one of many thousands of people who follow the same routine every day. His travel wastes few hours daily to travel earlier than his job's time to avoid traffic jams. Similarly, he reaches home late every day because of the traffic. He frequently faces trouble on the road because of the aggressive and few times sleepy drivers. He needs a solution to avoid the risk of accidents, reduces his number of hours wastage due to traffic congestions, helps to avoid the routes where traffic situations can get worse and consistently guides him on the road and weather conditions.

5.2 SIoV Use Case Scenarios

Prof. Adel drives a smart car that offers a range of different SIoV applications. The car ensures its safety by using many sensors deployed within the car. Furthermore, the car knows about the professor's social contact and has built its own social circle by establishing new relationships with other vehicles sharing the same interests. It also keeps the fog layer updated by consistently synchronizing its data and service information.

5.2.1 Safety and Point-of-interest applications

Prof. Adel was travelling towards his workplace. He carefully drove even though his mind is occupied with his meeting plans and tasks to be done. The safety application could notify because it had received a real-time over-speeding update from a service of a trust-worthy social contact with whom it had a short-term social relationship. He was around a mile away from his workplace when his car notified him that it's better change his route, because his usual entry gate of his University was closed for some work. This notification was based on the information that was sent by a long-term co-worker social relation who usually travels to the University around the same time.

5.2.2 Accident Warning application

Prof. Adel started his journey back to home from work. He had a long and busy day at work and wanted to reach home as soon as possible. Therefore, he decided to take a short route to home. As he changed his lane and was about to take his first exit towards his usual route to go back home, he was alerted by his smart car about an accident that just happened around 2 km away on that route. He checked the other route and started travelling towards that route. This accident warning application was working by consuming the fog based services and could notify the driver in real-time about the traffic that was few km away. The fog layer could notify the specific vehicle because it was a social contact of few vehicles that reported the information about the congestion. In this use case, the social circle of Prof. Adel's vehicle helped him to avoid delays in his journey.

The data generated by each person are reaching gigabytes and the trend is surging. However, smart vehicles in SIoV are expected to generate terabytes of data in a day from its various sensors, cameras and

social relationship updates. Moreover, the SIoV architecture will also include other transportation domains such as metros, trains, trucks, and flying vehicles. The fog-based distributed architecture becomes essential for the provision of services to enable real-time applications that require data in a relatively broader context compared to gateway layer.

5.2.3 Highway Information application

Prof. Adel was on his way to home, when suddenly he faced some issue with his car on the highway. He parked his car on a hard shoulder and checked his emergency mode that was already switched on by the smart vehicle. He used his smart vehicle interface to get an update and discovered that his car has already reported the incident to the system. He was also able to see that the help is half-a-hour away. However, the application also showed him the second option to start a video chat with the technical team who thinks that the issue can be easily fixed by the driver based on their analysis from the reported fault data. He started the chat and could fix the issue in a few minutes and continued his journey home. This use case depends on the cloud layer of the SIoV architecture. The vehicle could send a report of the incidence that eventually reached the cloud that analyzed the data. The cloud then shared the analyzed information with the concerned technical team that recommended both options. Finally, the communication link with the technical team was also facilitated by the cloud layer.

6. Future Research Challenges

The article assists in filling the gap in literature by outlining a scalable SIoV architecture that sets foundation for resolving issues like scalability, flexibility, decentralization, dynamicity, heterogeneity and context awareness. However, due to enormity of SIoV, several challenges are still to be investigated. This section provides an overview of future research challenges for SIoV systems.

6.1 Quality of Service (QoS)

Diverse nature of SIoV components makes it a significant challenge to meet assorted QoS requirements. Connected vehicles, RSUs and cloud services are expected to play a vital role in future ITS by gathering and processing enormous data at entity level. SIoV systems are hence required to accommodate numerous service requests with varied requirements. For example, safety services require low latency, high efficiency and reliable delivery. The QoS requirement of SIoV in safety applications is relatively stringent in terms of throughput and delay, due to real-time data information. Similarly, media sharing services in SIoV require high bandwidth and stable connection. Heterogenous nature of these services also pose a challenge when it comes to delay tolerance, reliability and efficiency. Such challenges require extensive research to ensure fulfilment of QoS requirements for varied services in SIoV.

6.2 Resource Management

SIoV components especially, vehicles are generally constrained by limited resources like, computation, processing, storage, and radio spectrum bandwidth that result in low data processing and computing capabilities. Due to complex computation, heavy processing and substantial storage demands of SIoV applications, including in-vehicle entertainment, vehicular social networking, internet sharing, and location-based services, it becomes increasingly challenging for a single entity of SIoV to efficiently support these applications. To overcome this challenge, resource management is required. However, due to dynamic and heterogeneous nature of SIoV systems, resource management can be challenging and can cause serious inconvenience if not handled appropriately. For example, safety applications require priority while scheduling resources based on the level of severity; this resource scheduling should be done with high efficiency and reliability to avoid unfortunate incidences. Substantial number, diverse nature, limited capabilities and high-

performance requirements of the resources in SIoV entails extensive research for development of a scalable and flexible framework for resource management.

6.3 Energy Efficiency

SIoV embodies several components that require energy for their operations e.g., vehicles, RSUs and cloud devices, etc. SIoV has an ever-increasing number of components that can enormously increase energy consumption for the overall system. The connected components in SIoV can significantly improve the scalability of the network, however, it aggravates the communication hopping that results in high overall energy consumption. Furthermore, dynamicity and decentralization in SIoV systems assist in empowering individual entities that gives them flexibility of socializing with other entities of the network, however, individuality of entities would result in processing, computing and storage of certain information at entity level that escalates energy consumption at entity end. Energy efficient solutions can greatly reduce the cost of SIoV systems however, extensive research is required to propose an energy efficient framework for a scalable and flexible system.

6.4 Security and Privacy

A reliable SIoV architecture requires the provision of end-to-end security and privacy for its successful adoption. In SIoV, the security threats which affect even a single vehicle can result in chaotic consequences by putting human lives at risk. The existing standardized security solutions are not designed for systems like SIoV. For example, a manufacturer can protect each smart part of a vehicle, but it will need future security updates to deal with the latest threats during its lifespan. The response to any security threat should be amicably dealt while ensuring the safety. For example, the detection of any malicious code in a moving vehicle should be fixed without any disruption to its usual operation. The privacy of users' information is also needed to be addressed by the SIoV architecture. Moreover, the management of trust between vehicles and infrastructure elements will be a key to ensure the security and privacy of SIoV.

7. Conclusion

SIoV is an emerging trend in the ITS domain that leverages the social aspect by focusing on interaction between different entities of the system. Vehicles, RSUs, passenger, drivers and pedestrians socialize with each other based on context, scale, and environment and information requirement. Design of SIoV architecture poses several challenges like decentralization, scalability, security and privacy, context-awareness, heterogeneity, dynamicity and interoperability. This article is an attempt to propose a scalable SIoV architecture based on Restful web technology to provide a foundation for developing SIoV applications, emphasize on the importance of web technology to meet the required interoperability for supporting the composition of diverse services and highlight the enabling technologies and protocols for SIoV systems. The article delivers an insight into the layered architecture of SIoV by illustrating the role and architecture of each entity of the system along with enabling technologies and protocols. Another main contribution of this article is to highlight the social relationships between different entities of the system along with the management of these relationships keeping in mind the dynamic nature of SIoV systems. Finally, the article analyzes the proposed SIoV architecture by demonstrating distinct use cases articulating the viability of the proposed architecture.

Acknowledgement

This work is supported by College of Computer Information Technology, American University in the Emirates, Dubai, United Arab Emirates and Hankuk University of Foreign Studies Research Fund of 2017 and National Research Foundation of Korea (2017R1C1B5017629).

References

[1] Gerla, M., Lee, E. K., Pau, G., & Lee, U. (2014, March). Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds. In Internet of Things (WF-IoT), 2014 IEEE World Forum on (pp. 241-246). IEEE.

[2] Nitti, M., Girau, R., Floris, A., & Atzori, L. (2014, May). On adding the social dimension to the internet of vehicles: Friendship and middleware. In Communications and Networking (BlackSeaCom), 2014 IEEE International Black Sea Conference on (pp. 134-138). IEEE.

[3] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future generation computer systems, 29(7), 1645-1660. *Elsevier*.

[4] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4), 2347-2376.

[5] Jara, A. J., Olivieri, A. C., Bocchi, Y., Jung, M., Kastner, W., & Skarmeta, A. F. (2014). Semantic web of things: an analysis of the application semantics for the iot moving towards the iot convergence. International Journal of Web and Grid Services, 10(2-3), 244-272.

[6] Atzori, L., Iera, A., Morabito, G., & Nitti, M. (2012). The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. Computer networks, 56(16), 3594-3608.

[7] Zheng, K., Zheng, Q., Chatzimisios, P., Xiang, W., & Zhou, Y. (2015). Heterogeneous vehicular networking: A survey on architecture, challenges, and solutions. IEEE communications surveys & tutorials, 17(4), 2377-2396.

[8] Bonomi, F. (2013). The smart and Connected Vehicle and the Internet of Things. In Invited Talk, Workshop on Synchronization in Telecommunication Systems.

[9] Wan, J., Zhang, D., Zhao, S., Yang, L., & Lloret, J. (2014). Context-aware vehicular cyber-physical systems with cloud support: architecture, challenges, and solutions. IEEE Communications Magazine, 52(8), 106-113.

[10] Gandotra, P., Jha, R. K., & Jain, S. (2017). A survey on device-to-device (D2D) communication: Architecture and security issues. Journal of Network and Computer Applications, 78, 9-29. *Elsevier*.

[11] Kaiwartya, O., Abdullah, A. H., Cao, Y., Altameem, A., Prasad, M., Lin, C. T., & Liu, X. (2016). Internet of vehicles: Motivation, layered architecture, network model, challenges, and future aspects. IEEE Access, 4, 5356-5373.

[12] Contreras, J., Zeadally, S., & Guerrero-Ibanez, J. A. (2017). Internet of Vehicles: Architecture, Protocols, and Security. IEEE Internet of Things Journal.

[13] Yang, F., Li, J., Lei, T., & Wang, S. (2017). Architecture and key technologies for Internet of Vehicles: a survey. Journal of Communications and Information Networks, 2(2), 1-17.

[14] Luan, T. H., Lu, R., Shen, X., & Bai, F. (2015). Social on the road: Enabling secure and efficient social networking on highways. IEEE Wireless Communications, 22(1), 44-51.

[15] Ning, Z., Xia, F., Ullah, N., Kong, X., & Hu, X. (2017). Vehicular social networks: Enabling smart mobility. IEEE Communications Magazine, 55(5), 16-55.

[16] Vegni, A. M., & Loscri, V. (2015). A survey on vehicular social networks. IEEE Communications Surveys & Tutorials, 17(4), 2397-2419.

[17] Mezghani, F., Dhaou, R., Nogueira, M., & Beylot, A. L. (2014). Content dissemination in vehicular social networks: taxonomy and user satisfaction. IEEE Communications Magazine, 52(12), 34-40.

[18] Maglaras, L. A., Al-Bayatti, A. H., He, Y., Wagner, I., & Janicke, H. (2016). Social internet of

vehicles for smart cities. Journal of Sensor and Actuator Networks, 5(1), 3.

[19] Alam, K. M., Saini, M., & El Saddik, A. (2015). Toward social internet of vehicles: Concept, architecture, and applications. IEEE access, 3, 343-357.

[20] Hou, X., Li, Y., Chen, M., Wu, D., Jin, D., & Chen, S. (2016). Vehicular fog computing: A viewpoint of vehicles as the infrastructures. IEEE Transactions on Vehicular Technology, 65(6), 3860-3873.

[21] Mahmud, R., Kotagiri, R., & Buyya, R. (2018). Fog computing: A taxonomy, survey and future directions. In Internet of Everything (pp. 103-130). Springer, Singapore.

[22] OpenFog Consortium Architecture Working Group. (2017). OpenFog Reference architecture for fog computing. OPFRA001, 20817, 162.

[23] Assunção, M. D., Calheiros, R. N., Bianchi, S., Netto, M. A., & Buyya, R. (2015). Big Data computing and clouds: Trends and future directions. Journal of Parallel and Distributed Computing, 79, 3-15.

[24] Stojmenovic, I., Wen, S., Huang, X., & Luan, H. (2016). An overview of fog computing and its security issues. Concurrency and Computation: Practice and Experience, 28(10), 2991-3005.

[25] Singh, S., Jeong, Y. S., & Park, J. H. (2016). A survey on cloud computing security: Issues, threats, and solutions. Journal of Network and Computer Applications, 75, 200-222.

Talal Ashraf Butt is an Assistant Professor at the American University in the Emirates. He holds a PhD in Internet of Things from Loughborough University, UK. He has also worked as a part of 5G Innovation Centre at the University of Surrey. Dr. Butt is a reviewer of IEEE journals and is passionate about next generation networks and protocols.

Razi Iqbal is an Associate Professor at the College of Computer Information Technology at the American University in the Emirates. Dr. Razi earned his PhD and Master's degree in Computer Science and Engineering from Akita University in Akita, Japan. He is currently a member of IEEE and IEEE computer and computational society.

Sayed Chhattan Shah is an Assistant Professor of Computer Science in the Department of Information Communication Engineering at Hankuk University of Foreign Studies Korea. He is also Director of Mobile Grid and Cloud Computing Laboratory. He received his Ph.D. in Computer Science from Korea University in 2012.

Tariq Umer received his Ph.D. in Communication systems in 2012 from School of Computing & Communications, Lancaster University, U.K and Masters in Computer Science in 1997 from Bahauudin Zakariya University, Multan, Pakistan. He has served the IT education sector in Pakistan for more than 13 years. He is the active member of Pakistan Computer Society and Internet Society Pakistan.