Enabling Smart Cloud Services Through Remote Sensing: An Internet of Everything Enabler

Sherif Abdelwahab, *Member, IEEE,* Bechir Hamdaoui, *Senior Member, IEEE,*
Mohsen Guizani, *Fellow, IEEE,* and Ammar Rayes

***Abstract*—The recent emergence and success of cloud-based services has empowered remote sensing and made it very possible. Cloud-assisted remote sensing (CARS) enables distributed sensory data collection, global resource and data sharing, remote and real-time data access, elastic resource provisioning and scaling, and pay-as-you-go pricing models. CARS has great potentials for enabling the so-called Internet of Everything (IoE), thereby promoting smart cloud services. In this paper, we survey CARS. First, we describe its benefits and capabilities through real-world applications. Second, we present a multilayer architecture of CARS by describing each layer's functionalities and responsibilities, as well as its interactions and interfaces with its upper and lower layers. Third, we discuss the sensing services models offered by CARS. Fourth, we discuss some popular commercial cloud platforms that have already been developed and deployed in recent years. Finally, we present and discuss major design requirements and challenges of CARS.**

***Index Terms*—Cloud computing, Internet of Everything (IoE), remote sensing, smart cloud services.**

I. Introduction

REMOTE sensing plays a key role in the acquisition of data about and from everything without needing physical field visits. Instead, it relies on sensory things or objects to sense and collect data remotely and in real time. With the recent emergence of cloud computing, remote sensing has been empowered and made possible even more than ever before. Unlike conventional ways of collecting and processing sensory data, cloud-assisted remote sensing (CARS) now enables: 1) *decentralization* of data sensing and collection, where sensory data can now be sensed and collected from everywhere instead of being restricted to limited areas; 2) *sharing* of information and cloud resources, where data and resources can be shared and used globally by all users; 3) *remote access* to global sensed information and its analytics, where sensed data can easily be accessed and analyzed online from everywhere; 4) *elastic provisioning* of cloud resources, where users can scale up and down their requested resources in real time based on demand; and 5) *pay-as-you-go* pricing models, where users can request (and hence pay for) only resources that they need based on their demand. CARS can then be viewed and foreseen as an emerging technology that has great potential for enabling the so-called Internet of Everything (IoE), thereby enabling smart cloud services.

IoE is a new Internet concept that tries to connect everything that can be connected to the Internet, where everything refers to people, cars, televisions (TVs), smart cameras, microwaves, sensors, and basically anything that has Internet-connection capability. A recent study by Cisco predicts that IoE is projected to create $14 trillion net-profit value, a combination of increased revenues and lowered costs, to private sector from 2013 to 2022 [1]. IoE is not seen as an individual stand-alone system, but as a globally integrated infrastructure with many applications and services [2].

When coupled with IoE, CARS gives birth to what we call Everything as a Service or Sensing as a Service (SenaaS) [3]. For example, a radio station can rely on CARS to lease a set of driverless cars which it can use to cover road accidents and broadcast traffic information [4]. This is becoming a commercially viable solution, especially after states such as Nevada, Florida, and California legalized driverless cars to operate in public roads [5]. Another example is to have traffic lights communicate with and control approaching vehicles upon sensing the presence of pedestrians and bikes [6]. Smart phones, which are typically (or can be) equipped with specialized sensors, can also be used to provide sensing and monitoring services for environment, healthcare, and transportation [7], [8]. These application domain examples are just a few among many others that can benefit from CARS to offer SenaaS.

SenaaS is a general term that describes a set of specialized sensing services that CARS can deliver. In addition to conventional data collection and processing services, CARS can provide virtualized sensing infrastructure as a service, as illustrated in the driverless car example before. CARS can also provide a development platform for remote sensing applications in which users do not own physical sensing resources, but can use them to develop and deploy their own applications.

In this paper, we survey the topic of using cloud computing to enable remote sensing services, referred to, throughout the paper, as CARS or simply CARS. First, we describe in Section II the benefits and uses of CARS by giving several real-world application domains where CARS can be applied. Second, we present in Section III a layered architecture for CARS that consists of four layers. For each layer, we describe the functionalities and responsibilities of the layer, as well as its interactions and interface with its upper and lower layers. Third, we discuss in

Manuscript received November 29, 2013; revised March 20, 2014; accepted May 05, 2014. Date of publication May 16, 2014; date of current version June 12, 2014. This work was supported in part by Cisco (CG-573228) and in part by the National Science Foundation (NSF) (CNS-1162296).

S. Abdelwahab and B. Hamdaoui are with the Electrical Engineering and Computer Science (EECS) Department, Oregon State University, Corvallis, OR 97331-5501 USA (e-mail: abdelwas@eecs.oregonstate.edu; hamdaoui@eecs.oregonstate.edu).

M. Guizani is with Qatar University, Doha, Qatar (e-mail: mguizani@ieee.org).

A. Rayes is with Cisco Systems, San Jose, CA 95134 USA (e-mail: rayes@cisco.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JIOT.2014.2325071

Section IV three different models of sensing services that CARS can offer: infrastructure-based, platform-based, and analytics-based models. Fourth, we go over in Section V popular commercial CARS platforms that have already been developed and been ready to use by highlighting their capabilities and their offered service models. Fifth, we present in Section VI architectural and functional design requirements of CARS, and discuss open research challenges and issues associated with each of these requirements. We conclude the paper in Section VII.

II. CARS Services and Applications

CARS can be found useful in many real-world applications in various domains, including industrial, medical, social, and environmental. Essentially, CARS services and applications extend traditional sensor network applications by enabling:

1) *distributed data collection and sensing*, where sensory information can be sensed and collected from everywhere;
2) *global data and resource sharing*, where sensory information and resources can be shared globally;
3) *remote and real-time data access,* where sensed data can be accessed and analyzed in real time from anywhere;
4) *elastic resource provisioning and scaling,* where service users can provision and scale up and down their needed resources based on demand;
5) *pay-as-you-go pricing,* where cloud users can request, release, and pay for resources whenever needed.

In this section, we classify CARS usages and benefits into three categories: 1) *remote tracking and monitoring*; 2) *real-time resource optimization and control*; and 3) *smart troubleshooting*, and present practical applications from each category and discuss how these applications benefit from CARS.

A. Remote Tracking and Monitoring

CARS enables remote tracking and monitoring of things of interest in real time, thereby allowing alerts to be raised and appropriate actions to be taken in a timely manner whenever needed. CARS can potentially be used for tracking and monitoring animal behaviors, moving-object locations (e.g., vehicles), environmental conditions, building surveillance and security, patient-health conditions, smart-grid operations, aviation and aerospace safety, and vegetation production quality, just to name a few.

1) Animal Behaviors: CARS can track and record locations of animals (livestock or wild) by collecting global positioning system (GPS) data in real time and storing it in the cloud. It can then provide an ecologist, for example, with an easy access to this location data, allowing the conduct of useful studies, such as studying migration activities, habitat, and grazing behaviors of various animals [9].

2) Environmental Condition Monitoring: CARS can be used to monitor environmental condition changes by having distributed sensors, which collect and send data to the cloud. Data to be collected can, for example, be measurements of water/air quality, temperature, humidity, pollution, carbon dioxide concentrations, etc. An environmental specialist can then access this data and use it to monitor climate changes, estimate concentrations of dangerous chemicals in water and air [10], monitor and alert of possible forest fires [11], and get alerted by possible avalanche occurrences [12].

3) Agricultural Monitoring: CARS has potential uses for monitoring vegetation remotely, combating corps-threatening pests, and improving farming system productivity. Smart camera technology is being used for remote vegetation monitoring by collecting and processing field images on the fly, generating performance reports, and sending processed reports to the cloud.[1] Reports can then be accessed also remotely and in real time by specialists to take appropriate actions to improve food security and farming system productivity [13].

4) Building Surveillance and Security: CARS can also be used to monitor and detect safety and security patterns, such as malicious human activities and building structural problems, allowing to raise security alerts in a timely manner. Smart cameras and motion sensors can be deployed in buildings and shopping malls to gather real-time data and send it to the cloud. For example, metal detectors are being used to feed data into a smart cloud application to allow the detection of explosives and terroristic attacks in real time [14], [15].

5) Healthcare Monitoring: Healthcare too requires an automated system that allows continuous monitoring of patients' health conditions and real-time reporting of abnormalities and condition alerts to physicians. Body temperature, blood pressure, and heart pulses are examples of things that can be sensed and sent to the cloud, where physicians can have easy, remote, and real-time access to. Wearable sensors are being used to track patient condition and location, develop contextual information, and alert against unusual situations [16]–[18].

6) Smart Meter Readings: CARS can be very handy in measuring and collecting smart grid operation information, allowing easy integration and automation of smart grid components. Smart meters, for example, are being used to measure electrical energy usage remotely and report these measurements to the cloud, allowing real-time monitoring and analysis [19].

7) Aviation and Aerospace Safety: CARS can be used to improve security and safety of crew and air vehicles. Sensed data from different wing sensors, wearable sensors by crew, leakage detectors, and pressure sensors can be collected and sent to the cloud in real time, where it can be used in ground testing and space vehicle health monitoring to ensure crew and vehicle safety [20].

B. Real-Time Resource Optimization and Control

CARS can also enable real-time optimization and control of various resources, where resources vary from one application domain to another. Waste management, traffic control, smart parking, water/irrigation management, and guided navigation are a few applications where resources can be optimized and controlled efficiently via CARS. Depending on the application, CARS can then reduce operational costs, augment crops productivity, and improve system reliability and performance.

[1] [Online]. Available: http://research. engr. oregonstate. edu/ecomonitoring/

1) Waste Management: CARS allows remote monitoring of recycling containers, making recycling waste pickup and routing processes effective. Waste management is a complex process that involves generation, on-site handling, collection, transfer, processing, and disposal of waste. By connecting various elements of this process and feeding data into the cloud, on-site handling efficiency, storage, and transfer can be greatly improved [21]. A thoughtful analysis of this data supplies authorities with adequate information to act upon to improve waste management efficiency.

2) Smart Parking: Using CARS, one can provide real-time information about parking availability to locate parking spots faster and more efficiently, which saves time of car drivers and reduces energy consumption. This can be achieved by quickly identifying and reserving a free parking spot in the vicinity of the drivers' final destination [22].

3) Traffic Control: CARS can also be very useful in traffic monitoring and control. It enables quick collection of real-time traffic information that can be used to avoid traffic jams, report accidents, report road constructions, and reduce fuel consumption. Traffic information can be sensed and collected by designated sensors, or by participatory sensing devices (e.g., smart phones and smart cameras) to monitor roads and traffic conditions in cities [23]. Not only can the collected data be utilized for traffic monitoring, but also for traffic routing and speed tracking [18], [24]. In addition, CARS can also be used to manage fleets in public and private transportation [25].

4) Healthcare: CARS can be used in healthcare systems to study and control disease transmission, and support independent living of elderly, inferior, and people with disabilities. In [26], sensors are used to construct a social network of high school students to control disease transmission. The system developed in [27] proactively informs visually impaired people with possible risks. Mobile phones and specialized wearable sensor networks are being utilized to detect elderly people body posture and biological signals to enable efficient allocation of needed clinical resources for diagnosis, treatments, and surgeries [28], [29].

C. Smart Troubleshooting: Identify, Diagnose, and Repair

CARS makes remote problem identification, diagnosis and repairing possible and efficient. Application domains where this technology can be applied to are diverse; aviation and aerospace, automotive, network systems, buildings, smart grids, and oil and gas pipelines are some examples.

1) Aviation and Aerospace: CARS can remotely provide pilots, dispatchers, and space mission controllers with necessary and useful information that help them quickly identify, locate, and fix problems. For example, it can allow pilots and dispatchers to avoid hazards of in-flight icing, and allow engine producers to detect cracked fan blades and other malfunctioning engine components remotely and fast [30].

2) Automotive: This industry has been witnessing a paradigm shift, as makers have been installing hundreds to thousands of sensors and embedded systems in their vehicles. One key use of these sensors is for enabling smart and efficient maintenance. As an example, Hull *et al.* [31] develop a computing system that takes advantages of these deployed sensors to collect, process, and visualize sensed data to enable remote and in-vehicle preventive diagnosis. CARS can improve maintenance in automotive industry even further by making this sensory data available in the cloud and be remotely accessible to experts in real time so that diagnosis and necessary repairs can be made fast and effectively.

3) Network Systems: CARS provides networking and information technology (IT) troubleshooters with the ability to remotely diagnose and identify networking component failures, and repair them also remotely and in a timely manner. For example, CARS has been used for providing real-time monitoring of data cloud centers to identify and repair (hardware and software) faults [32]–[34] and to make wise energy management decisions so as to reduce energy consumption in data centers [35], [36].

4) Buildings: CARS can render buildings and bridges act smarter. It can enable easy and fast detection and localization of light failures, air conditions stalling, or structural defects. For example, it can help sense ambient structural vibrations to allow fast detection of structural problems and hazards in buildings, bridges, and any complex structures [37]–[39]. As another example, CARS can assist in reducing energy consumption in buildings [40], [41].

5) Smart Grids: Power grids are becoming very complex in size, heterogeneity, and behavior. CARS can play an extremely critical role in keeping these grids operational at minimum costs. One potential use of CARS is to help maintain these grids fault-free for as long as possible by enabling remote failure detection and diagnosis so that repairs can be made quickly [42]. Connecting generation units, transformers, transmission lines, motors, and other grid components to the cloud and having them send their sensed data directly to the cloud enable grid operators to remotely analyze system behaviors in real time, allowing the detection and fixing of system faults in a timely manner.

6) Oil and Gas Pipelines: These rely heavily on fast failure detection typically enabled through monitoring of special parameter values (e.g., leakage and high pressure) collected via designated sensors that are spread out along the pipelines (length could be thousands of kilometers). CARS can make data monitoring, collection, and analysis more efficient than ever, thus reducing pipeline maintenance cost and increasing pipeline productivity [43], [44].

III. CARS Architecture for Enabling IoE

The discussion of the different CARS application domains in Section II calls for a new architectural design of CARS in order to be able to enable smart cloud services. This motivation is triggered by the unique characteristics and features that distinguish CARS from other traditional distributed systems. Sensory infrastructure deployment and sensing technique development across different domains may share common challenges and specificities, which should be taken into account when designing an architecture. Another feature that CARS architecture needs to support is resource sharing across different domains. That is, sensors and/or sensory infrastructures belonging to one domain should be shared and used by another domain when needed. For

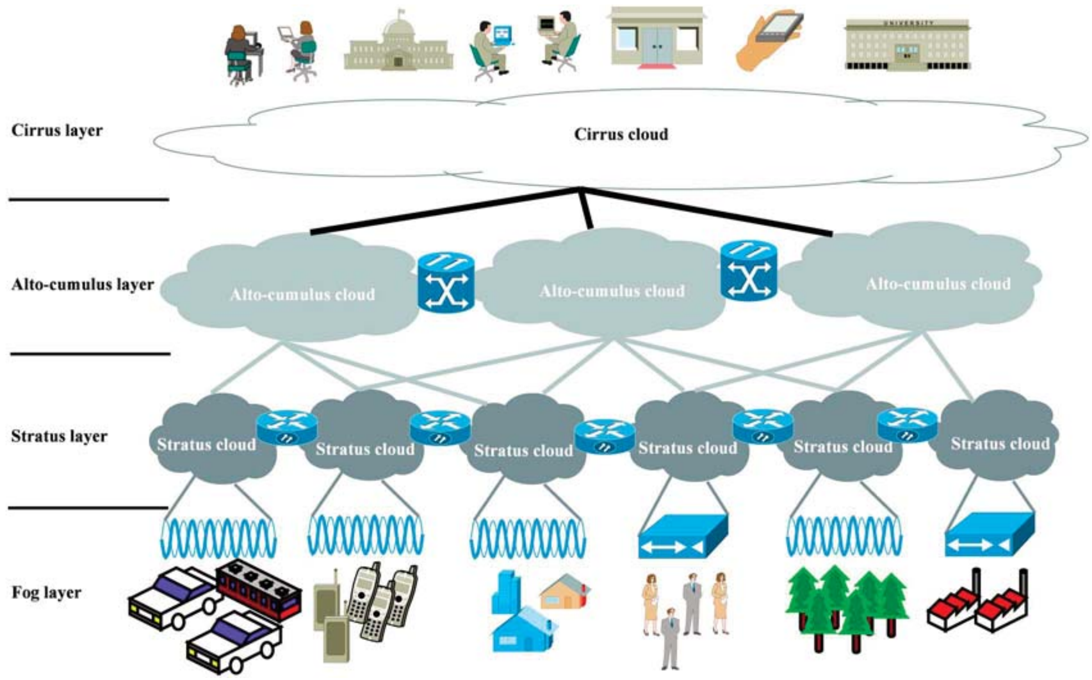

Fig. 1. Proposed CARS architecture.

example, it should be possible for a smart camera deployed in a mall for security and surveillance purposes to also be used for analyzing and studying market and customer behavior trends.

In this section, we present the proposed CARS architecture. Such an architecture can be viewed as a geographically distributed platform that connects many billions of sensors and things, and provides multitier layers of abstraction of sensors and sensor networks. Fig. 1 shows the proposed CARS architecture, which has four main layers: 1) fog layer; 2) stratus layer; 3) alto-cumulus layer; and 4) cirrus layer. Layer roles and inter-layer interactions are described in detail as follows.

A. Fog Layer

Fog layer encapsulates all physical objects, machines, and anything that is equipped with computing, storage, networking, sensing, and/or actuating resources, and that can connect to and be part of the Internet. The sensory elements of this layer are those that collect and send raw sensed data to stratus layer, by either being pulled by stratus layer or being pushed by fog layer to stratus layer. Major functions of fog layer are to provide:

1) heterogeneous networking and communication infrastructures to connect billions of things;
2) unique identification of all things through Internet Protocol Version 6 (IPv6);
3) data aggregation points to serve as sensing clusters.

Although the capabilities of sensing elements of this layer may vary depending on their type and purpose, it is safe to consider that they are very likely to be resource (e.g., power, memory, and CPU) limited and that they may operate on opportunistic wireless access mode only. As a result, sensor clustering and network virtualization techniques have emerged as possible ways to tackle such resource limitation of sensory devices, thus enabling distributed CARS services. But this is a cumbersome task due to the huge numbers of sensors and to their heterogeneity. Classification and virtualization of sensing and actuating networks (SANs) is a multidimensional problem, and can, for example, be: context-based (e.g., manufacturing, urban planning, and healthcare), location-based (e.g., ocean and urban area), ownership-based (e.g., personal, public, and participatory), or mobility-based (e.g., static and mobile). When mobility is considered as the metric for categorization, we have the following three types:

1) *fixed SANs*, where sensors are stationary, and network connectivity is known and controllable;
2) *mobile SANs*, where sensors are mobile, but their locations are known and controllable (e.g., Google driverless vehicles), and network connectivity is known and may be controllable;
3) *participatory SANs*, where sensors are private devices (e.g., smart phones) owned by people who, at their will, may or may not choose to participate in sensing tasks [7], [8]. Network connectivity could be intermittent and may or may not be known, but definitely not controllable.

B. Stratus Layer

Stratus layer is a mid, tier-2 layer that consists of thousands of clouds whose main resources are sensory devices and SANs. Each stratus cloud manages and acts as a liaison for a different group of SANs that share similar features, context, or properties. The functions of stratus layer include:

1) abstracting and virtualizing physical SANs through virtual network embedding (VNE) techniques;
2) handling and managing virtual SAN migration and portability across different clouds;
3) managing and ensuring operations and functionalities of virtual SAN instances;

4) enabling and managing (physical or virtual) SAN configurations to ensure network connectivity and coverage;
5) controlling the layer's operations and functionalities to ensure that customers' service level agreements (SLAs) communicated from higher layers (as detailed later) are met.

Stratus layer provides an abstraction of the physical world represented via fog layer to alto-cumulus layer. This layer does not interact directly with CARS customers, but serves them through requests received from higher layers. Conceptually, stratus clouds are optimized to handle heterogeneous connectivity of SANs by having software-defined networking interfaces with alto-cumulus layer. Several commercial solutions that currently exist can be considered as stratus clouds, as discussed in Section V.

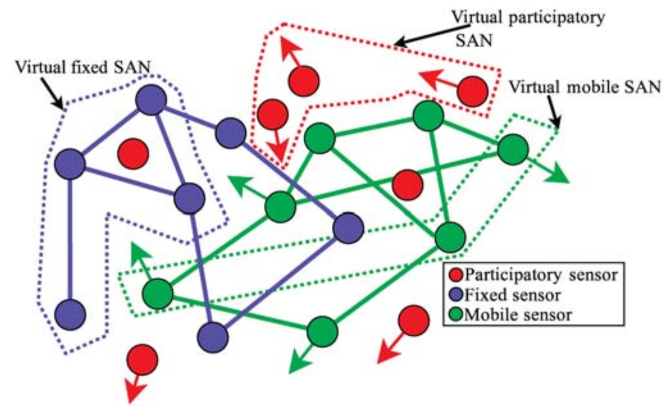

Fig. 2. Virtual SAN embedding.

C. Alto-Cumulus Layer

Alto-cumulus is a middle layer that serves as a point of liaison between stratus and cirrus layers. It facilitates negotiations related to pricing, policy and regulations, and SLAs between stratus and cirrus layers, and ensures that the agreed upon terms are not violated. While stratus clouds are domain specific; i.e., each cloud is very likely to be concerned with one application domain (e.g., medical, environment, and agriculture), an alto-cumulus cloud may map to and orchestrate multiple stratus clouds belonging to different domains. As mentioned earlier, this can enable inter-cloud resource sharing, thereby increasing resource elasticity and scaling. Major functions of alto-cumulus are as follows:

1) serving as a point of liaison between cirrus and stratus layers by translating policy and regulation requirements expressed by cirrus layer into domain-specific requirements understood by stratus;
2) enabling business and payment transactions between cirrus and stratus layers by providing two-way brokerage services;
3) enabling and facilitating SLA negotiations between cirrus and stratus, and monitoring and ensuring that these SLAs are met; i.e., playing the role of a policy enforcement agent;
4) coordinating and facilitating inter-cloud interactions, data exchange, task migration, and resource sharing across different stratus clouds.

D. Cirrus Layer

Cirrus layer is the highest layer in the CARS architecture, and its main role is to interact with CARS service customers and satisfy their requests with the aid of lower layers. It does not deal with resource virtualization, nor does it need to know which cloud handles which resources. It just needs to communicate customers' requests specified via their SLAs to alto-cumulus clouds. The major functions of this layer are as follows:

1) acting as the customers' entry point to CARS systems by allowing them to specify their required services via SLAs and to select their desired service models;
2) allowing CARS customers to set up their sensing task requirements and do whatever their chosen service model allows them to do (e.g., software configuration/deployment);
3) negotiating SLAs with customers and communicating them to alto-cumulus layer;
4) providing online applications for remote data analysis to be used by customers to visualize their data in real time.

This layered CARS architecture essentially consists of connecting sensory devices (fixed, mobile, and participatory) through the fog layer, managing SAN virtualization and embedding through stratus layers, managing cloud domains and these virtual SANs through the alto-cumulus layer, and providing abstractions of cloud services to customers through the cirrus layer.

IV. CARS Service Models

The objectives of CARS are to provide cloud customers with flexible access to data and sensing services, allow them to develop their own domain-specific applications, and allow clouds to share physical resources. We classify the services offered by CARS into three smart service models, analogous to cloud computing's cloud service models: 1) IaaS; 2) PaaS; and 3) SaaS. The models are Sensing and Actuating Infrastructure as a Service (SAIaaS), Sensing and Actuating Platform as a Service (SAPaaS), and Sensing Data and Analytics as a Service (SDAaaS).

A. SAIaaS

SAIaaS model requires that physical sensor and SAN resources serve multiple sensing tasks concurrently. Customers cannot make changes to physical resources (i.e., SANs and sensors), but have full control over their allocated virtual instances. An example of this is investigated in [4], [5], where the authors propose to use Google driverless cars to provide sensing services by creating a mobile virtual network of cars. The cloud platform in this example receives a request for performing a complex sensing task. The request specifies some level of required connectivity of these mobile cars, and grants the request by creating and providing a virtual instance of the sensing/driverless cars. Fig. 2 depicts an example of multiple virtual SANs. In this illustration, three sensing tasks are requested to be

TABLE I
COMMERCIAL PLATFORMS SPECIALIZED IN CARS AND THEIR CATEGORIZATION IN TERMS OF CLOUD LAYER TYPE AND SERVICE MODELS

Platform	SAIaaS	SDAaaS	SAPaaS	Sensor connectivity	Infrastructure
Carriots [46]		√	√		Participatory
Coversant (SoapBox) [47]			√		Commercial devices
Digi (Now Etherois) [48]		√		√	Participatory
ILS Technology [49]		√	√	√	Industrial devices
Layer 7 [50]			√	√	Industrial devices
Numerex [51]	√	√	√	√	Cellular, satellite, WiFi, and short range devices
Arm [52]				√	Arm embedded microprocessors
Ubidots [53]		√			Participatory Raspberry PI and Arduino
Living PlanIT [54]		√	√		Smart cities and urban development
Arkessa [55]			√	√	Industrial and Arekessa devices
ThingWorx [56]			√	√	Participatory and industrial devices
Axeda [57]		√		√	Industrial devices
Maingate [58]		√			Smart Home, Smart Enterprise and Smart Grid
Xivily [59]			√		Participatory
Nimbits [60]		√		√	Participatory
ThingSpeak [61]			√		Participatory
Paraimpu [62]			√	√	Participatory
SensorCloud [63]			√		Participatory
Oracle [64]		√	√		Participatory and industrial devices

carried out by the platform, leading to the creation of three virtual SANs (two mobile and one fixed).

B. SAPaaS

SAPaaS model offers sensing platforms as a service. In this model, CARS customers are provided with a set of application programming interfaces (APIs) and libraries that they can use to develop their own sensing and actuating applications without worrying about the physical (SANs). For example, in healthcare applications, CARS service providers can allow customers to use certain APIs and libraries to manage data granularity, or to set parameters in sensors worn by patients, or to develop applications that perform data analysis [45]. These CARS customers, in this case, do have full control over their applications and can manage their resources, but they cannot alter any change in the physical or virtual infrastructure (e.g., shutdown a sensor, connect to another sensor, enable or deploy a new sensor, and change firmware or software).

C. SDAaaS

Many practical applications need only to have access and be able to process sensed data without needing to change anything in the physical sensors or in the virtual realizations of SANs. CARS service customers using this service model, referred to as the SDAaaS model, are only interested in the context in which sensed data is collected, its accuracy, and its sufficiency to be able to generate meaningful inferences. SDAaaS model is used, for example, in [32] to enable remote maintenance and troubleshooting of cloud data centers. In this example, sensed data is collected using SANs and sent to the cloud, where network system analysts use it to make resource management and optimization decisions. In order to provide maintenance service to their clients/users, cloud system analysts (who are here the CARS service customers) do not need to know (nor do they have control over) how data is collected, how sensors are deployed, or how softwares are setup. All what they require is that data should be available and accurate.

V. IoE Reality: Real-World Cases

Cloud computing and IoE industry has been gaining some momentum in recent years, and we have been witnessing the emergence of many cloud-based commercial services. The service offered so far by these commercial clouds is mainly twofold. First, it consists of allowing cloud service customers to attach their sensors to the platform. These sensors can then collect data and send it to the cloud. Then, it consists of allowing these customers to develop and install their own custom-made softwares to make use of their sensed data in anyway they see fit. Based on their supported functionalities, these commercial clouds can essentially be viewed as stratus clouds, when categorized according to the proposed layered architecture. We also observe that none of these commercial clouds/platforms support SAN virtualization, data or resource sharing among different clouds, or data and virtual instance migration across different clouds. These platforms do, however, support cirrus cloud functionalities by acting as a service entry to CARS service customers.

As far as service models are concerned, each commercial cloud uses one or a combination of SAIaaS, SAPaaS, and SDAaaS models. Some clouds extend their services to also enable (physical or virtual) connectivity among sensors [via machine to machine (M2M) communication]. Table I provides a comprehensive overview of these commercial clouds as examples of CARS providers, and a classification of their service models and of weather they support sensor connectivity.

TABLE II
SUMMARY OF DESIGN REQUIREMENTS AND RESEARCH CHALLENGES

Category	Requirements	Research Challenges	Needed Solutions/Mechanisms
Resource allocation and management	• Resource management mechanisms tailored to each layer function. • Availability- and mobility-centric resource management techniques.	Sensor availability and mobility	Develop resource management mechanisms that account for the limitations of sensory devices and the requirements of advanced sensing tasks.
		Customer-provider resource allocation	Develop tools and algorithms that allow to select efficient customer-cloud paths.
		Troubleshooting	Design methods that facilitate maintenance and troubleshooting during system operation.
		Architectural considerations	Develop architectures that allow interface interaction, define component functionality, and enable policy and preference specification.
Resource dimensioning and scaling	• Support of elastic resource provisioning. • Workload prediction techniques to make efficient resource provisioning decisions. • Effective pricing schemes. • Sensory device search, selection and ranking algorithms.	Cross-vendor resource sharing	Develop mechanisms that allow application scaling across multiple CARS providers.
		Pricing	Design fair pricing schemes that promote distributed and remote sensing.
		Workload characterization and prediction	Develop techniques that can characterize and predict sensing tasks.
		Sensing coverage and connectivity	Design resource dimensioning and scaling schemes that account for sensory devices coverage and connectivity.
SLAs and QoS	• Layer specific resource allocation and control techniques to meet defined and agreed upon SLAs. • Availability- and mobility-centric SLA definition and satisfaction.	Cross-layer awareness	Develop new cross-layer resource allocation mechanisms that supports SLAs.
		Admission control	Develop admission control mechanisms that enable Internet-based cloud providers to meet SLAs.
		Assessment tools	Develop new tools that can measure and assess SLAs and QoS metrics.
		Selection mechanisms	Create selection and coordination mechanisms that play the role of a point of liaison between CARS providers and their customers.
Sensor network virtualization and embedding	• Virtualization techniques involving distributed sensory platforms. • Embedding techniques of sensory platforms while accounting for practical aspects (e.g. connectivity, QoS, availability, and mobility). • Incentive mechanisms to promote resource pooling of participatory and private sensors.	Enabling embedding decentralization	Develop decentralized virtual resource embedding algorithms that support multiple objectives and enable smooth resource migration.
		Addressing environment heterogeneity	Develop embedding algorithms that tackle wireless related issues, such as mobility, lack of centralization, link instability, etc.
		Increasing sensor availability and reliability	Develop availability and reliability models that take into account the random/stochastic presence nature of sensory devices.
		Providing good incentive mechanisms	Develop incentive mechanisms that increase the likelihood of sensory device participation.
Intercloud interoperability and portability	• Cloud-to-cloud communication and intercloud interoperability protocols that require minimal changes to existing software, minimal interaction with cloud providers, and minimal input from customers. • Inter-layer resource sharing policies and agreements.	Intercloud migration performance metrics	Develop metrics that can be used to trigger intercloud migration.
		Assessment algorithms of cloud services performance	Develop protocols and tools that can be used to assess the cloud services performance metrics.
		Optimization algorithms of intercloud migration process	Develop protocols and tools that effectively select the best CARS provider on migration.
		Standards and ontologies	Establish unified APIs and standards for cloud computing incorporating CARS services.

VI. Architectural and Functional Design Elements: Requirements and Challenges

In this section, we present architectural and functional design components required for enabling CARS, and discuss some of the challenges associated with them. These requirements are classified into: resource allocation and management, resource dimensioning and scalability, SLA and QoS, sensor network virtualization and embedding, and intercloud interoperability and portability. For convenience, we also provide in Table II a summary of all these design requirements, the research challenges associated with them, and some needed solution mechanisms.

A. *Resource Allocation and Management*

In general, resource management mechanisms vary depending on which layer they are designed for. According to the layers' functionalities of the architecture discussed in Section III, the resources that cirrus layer has control over and can manage consist mainly of classical cloud resources, such as data center servers, storage units, and networking resources. Cirrus clouds do not manage sensory devices or SANs, but they can share these resources by having access to virtual instances mapped to them through alto-cumulus layer. Alto-cumulus clouds deal with two types of resources: classical cloud resources and virtual SAN instances abstracted by the stratus layer. Stratus clouds, on the other hand, are only responsible for managing physical sensory resources (i.e., sensors and SANs). Whenever a request is received from an alto-cumulus cloud, stratus clouds grant the request by allocating virtual instances of sensors and/or SANs that satisfy the requested functional constraints and SLAs. Multiple virtual instances may share some or all of their physical sensory resources. For example, a stratus cloud may pool a physical SAN that can concurrently serve an airline operator and a fire-fighting station in the same city through two separate virtual instances. Also, an alto-cumulus cloud may pool multiple virtual instances from different stratus clouds.

Most existing works on resource management for cloud computing focus on server, storage, and networking resources, that is, resources that are mostly handled by the circus layer. Server virtualization (e.g., VMware and Xen[2]) has emerged as a key technology for improving cluster/cloud computing performances by allowing multiple virtual machines (VMs) to share and use the same physical machine (PM). As a result, it has been the research focus of many researchers during this past decade [65]–[70]. Inspired by this server virtualization concept, network virtualization has then recently emerged as a potential technology for enabling efficient resource management in CARS systems [71], [72]. When coupled with (SDN) [73], network virtualization provides a greater resource management flexibility in terms of not only creating and allocating virtual computing resources, but also interconnecting VMs, thereby enabling VNE.

1) Research Challenges: Resource management issues in CARS are far from being solved. Some of these research challenges are as follows.

a) Sensor availability and mobility: One source of challenges specific to CARS is the nature and characteristics of the sensory devices (i.e., sensors, actuators, and SANs). In general, these devices have very limited computing and energy resources, and may only have opportunistic network access [74]. Their locations and trajectories may not be known or even predictable, making it too difficult to build robust and reliable physical SANs. It could even be impossible to perform a task; e.g., when the sensing task is to report temperatures in a location where no sensors are available, this task cannot be performed. Some applications may even request that sensing be performed along some specific trajectory/path, requiring sensors to follow the specified path in order to perform the requested task successfully [75]. The nature of sensory devices, which constitute an important set of CARS resources, gives rise to new resource management issues that need to be addressed.

b) Customer-provider resource allocation: There has not been much research focus on developing selection algorithms that can find best customers-cloud paths. Path selection algorithms should consider performance parameters like reliability, bandwidth, connectivity, etc. In addition to optimizing path selection, these algorithms can be used for balancing customer request loads to optimize cloud performance.

c) Troubleshooting: In traditional networking paradigms, IT teams can easily identify and diagnose failures and performance related problems. Now due to the complexity and to the large number of components involved in such systems, it can be too difficult to troubleshoot technical problems. When designing these systems, it is important to make sure that maintenance procedures are incorporated at an early design stage so that maintenance and troubleshooting can be easily done during system operation.

d) Architectural considerations: There is a need for a comprehensive architecture that identifies the different entities and components of CARS systems, defines their functionalities and responsibilities, specifies their interfaces and interactions, and gives different entities a means to specify regulatory policies and express requirements and preferences.

B. Resource Dimensioning and Scaling

Supporting elastic resource provisioning, where service users can provision and scale up and down their needed resources based on demand, is one of cloud computing paradigm's key capabilities and features that led to its success. As such, CARS resources, such as sensors, (SANs), servers, and storage units, should be expandable in real time in order to accommodate for any demand sizes (e.g., those that arise during peak hours and during special events). CARS must enable this resource provisioning elasticity so that cloud resources appear boundless to CARS customers. One of the advantages of having layered CARS architecture is that it makes it easy to design scalable resource allocation techniques. Because of this layered architecture, the entire fog layer, possibly consisting of billions of sensory devices, can, for example, be abstracted, so that these devices appear to upper layers as resource pools.

Prediction techniques have been proposed as effective ways to improve resource provisioning decisions by anticipating workloads of various traffic types in advance [76]. This allows CARS service providers to shutdown or hibernate resources when not needed, thus reducing energy consumption as well.[3] These prediction algorithms can also be used to define minimal required resources, allowing to satisfy customers with different QoS requirements [70]. There have been many techniques proposed in literature to predict workloads, but mostly in the context of distributed (e.g., Grid) systems. Examples of these prediction techniques are: hidden Markov models [78], polynomial fitting [79], and hybrid models [80]. There have also been techniques proposed for managing cloud data center resources [81], [82]. These rely on machine learning approaches, such as neural networks, to predict the number of requests that could arrive at cloud-hosted web servers, allowing to turn off unneeded parts of the resources and thus saves energy.

Coming up with the right pricing models is also critical for promoting resource dimensioning and scaling [83], [84]. Existing models for cloud computing are mostly on a per-reservation basis, where CARS service customers reserve resources for some time, and pay for them regardless of whether they have or have not been used fully. Although this pricing approach may not the best option for CARS service customers, it eases resource dimensioning and pricing logistics for CARS providers.

[2][Online]. Available: Vmware–http://www.vmware.com/, Xen–http://www.xenproject.org/

[3]Accountable roughly for about 53% of total datacenter operational cost [77].

In a network constituted of billions of sensory devices, finding and locating the right sensors and/or SANs to perform specific sensing tasks is very challenging. Search, selection, and ranking algorithms that locate and design appropriate SANs need to be developed. Different algorithms are to be developed for different cloud layers, which depend on the functionalities that are specific to the layers (e.g., an alto-cumulus cloud needs to search for suitable stratus clouds while a stratus cloud needs to search for physical SANs). As an example, Perera *et al.* [85] developed an algorithm that searches for, selects, and ranks sensory devices based on CARS customers' expectations and priorities. The algorithm clusters sensors with similar functionalities.

1) Research challenges: This area of research is still immature and needs future focus. The following are some possible challenges.

a) Cross-vendor resource sharing: There is a need for resource sharing and scaling across different CARS providers, as it will not be possible for a single provider to have enough resources to accommodate for any demand size. It is therefore important to have mechanisms that enable application scaling across multiple CARS providers.

b) Pricing: While the pay-as-you-go pricing models can be a good model for easing pricing logistics between CARS providers and their customers, it may not promote resource dimensioning and scaling. Developing pricing schemes that facilitate resource dimensioning and elasticity remains an explored area of research that needs a thorough investigation. What makes this task even more challenging is that we are here dealing with not only classical types of cloud resources (such as computing, storage, and networking resources), but also sensory devices, giving rise to other types of challenges, such as mobility, availability, and willingness to participate. Recall that CARS relies heavily on distributed sensors to collect data, and even though most CARS vendors will probably be deploying and relying on their own sensory devices, reliance on participatory and private sensors to perform distributed sensory tasks cannot be avoided. Therefore, pricing schemes need to be designed with a goal of promoting distributed sensing by providing good incentives for private sensors to participate in CARS systems.

c) Workload characterization and prediction: Another way to improve resource scaling is to rely on prediction algorithms to determine and estimate cloud service workloads in advance. Some research efforts have already been put to develop workload prediction techniques for cloud computing, but mostly deal with cloud data centers with the objectives of balancing workloads across servers, increasing servers' utilization, and/or reducing power consumption. However, not much has been done when it comes to cloud-based remote sensing. Techniques that can characterise and predict sensing tasks (which depend on location, sensing domains, etc.), sensory device availability, SAN connectivity and coverage have not been investigated yet. Generally speaking, prediction for remote sensing is an unexplored research area.

d) Sensing coverage and connectivity: One challenge lies in how to maintain and ensure that an area of interest to be sensed is covered fully. Challenges arise especially when some of the physical nodes are participator sensors which are often mobile, have limited computing and energy resources, and have unpredictable behaviors.

C. *SLAs and QoS*

Another CARS key requirement is the ability to allocate and control resources so as to meet SLAs. This requires resource usage monitoring and admission control mechanisms that guarantee fulfillment of contracted SLAs, as well as protocols and procedures that report resource usage to CARS service providers and customers. There is a clear split of roles among cloud layers when it comes to handling SLAs. Cirrus clouds are, for instance, responsible for defining and negotiating SLAs, because of their direct interface with customers, while alto-cumulus and stratus clouds are responsible for ensuring that SLAs are met.

In cloud computing, SLAs are viewed as contractual obligation metrics introduced to define the minimum required service levels and the resolution policies to be followed when CARS providers fail to satisfy SLAs [86]. Examples of SLAs include VM availability, CPU performance, recovery expectation, data format, and diagnosis and repairing procedures.

Being able to guarantee a certain level of reliability is fundamental to ensuring the fulfillment of contracted SLAs. In an effort to do so, Pezoa and Hayat [87] develop models that capture the impact of stochastic and correlated failures on cloud reliability. These models can be used to develop dynamic task re-allocation policies while maximizing reliability of data cloud centers. In the context of (SANs), reliability is defined as the probability that every point in the sensed field can be observed by at least K nodes and that there exists at least one operational path from each of these K nodes to the sink node [88]. Coverage, on the other hand, is viewed as setting K to one in the reliability definition above [89].

Goudarzi and Pedram [66] propose multidimensional resource allocation schemes for data cloud centers with inter-server communication capabilities, and derive a closed-form formula of the average response time for multitier applications. They also derive an upper bound on the profit that can be achieved for given combinations of customers, resources, and SLAs. They introduce two SLA classes: 1) gold SLAs which consist of average response time target, maximal client arrival rate, a rewarding scheme for serviced request, and a penalty for missing response time targets and 2) bronze SLAs which consist of maximal client arrival rate only. There have also been some attempts that aim to provide resource scheduling and admission control schemes with the objectives of reducing energy consumption [70], meeting multiple types of SLAs [90], and providing efficient content storage [91].

1) Research Challenges: We discuss four challenging design requirements: cross-layer awareness, admission control, SLA assessment, and service provider selection.

a) Cross-layer awareness: New resource allocation mechanisms, taking into account cross-layer aspects, need to be developed to support SLAs. Existing techniques are to be revisited to ensure easy and smooth integration of SLAs aspects. Challenges arise mainly from cloud resource virtualization, geographic sensor distribution, sensor mobility and availability, pricing models and incentive mechanisms, and SLA negotiation procedures.

b) Admission control: Admission control has always been a big challenge when it comes to using and relying on the best-effort service model being adopted by the Internet. Because all of these cloud-based services reply on the Internet infrastructure to deliver their services to their customers, CARS necessitates the development of admission control algorithms to be able to guarantee SLAs. Admission control challenges still remain unsolved.

c) Assessment tools: It is important to have tools and protocols that can measure and assess SLAs and QoS metrics offered to customers. This helps CARS service providers make sure that the contracted SLAs and QoS obligations promised to customers are met. Not much has been done in this regard.

d) Selection mechanisms: We need to have cloud service provider selection mechanisms that allow service customers to select the CARS service provider that can meet their SLAs. One possible way of enabling this is by having an entity coordinate and play the role of a point of liaison between CARS providers and their customers. Little attention has been given to this provider selection topic.

D. Sensor Network Virtualization and Embedding

Network resource virtualization, also known as VNE, is envisioned to play an essential role in enabling distributed sensory platforms. It can act as an intermediate agent between infrastructure providers (e.g., SAIaaS providers) and service providers (e.g., SDAaaS providers) by decoupling their roles into one that deals with the deployment and maintenance of physical network components (links and nodes) and one that deals with deployment of software and protocols and with offering services to customers. It can enable efficient usage of network resources by mapping multiple different virtual instances to the same physical resources. It can also provide service providers with flexible and dynamic management of their virtual networks (change physical nodes and add/remote virtual links), allowing these providers to accommodate for unexpected service demand changes and to choose/replace their infrastructure providers with minimal efforts.

There have been some efforts aimed to enable virtualization of SANs by addressing various practical aspects, such as connectivity [92], QoS [93], and mobility [4], [75]. Alam *et al.* [92] propose a framework to support virtualization of event-driven sensors. The proposed framework addresses connectivity of different types of sensors through service-oriented architecture (SOA) techniques. In [94], Distefano *et al.* discuss management, abstraction, and virtualization of sensors, but not those of networks of sensors, such as SANs. In [93], Khan *et al.* propose a multilayered architecture that satisfies a set of requirements, concurrent task execution, sensor node clustering, and task prioritization, in virtual wireless sensor networks. In [95], Bandara *et al.* propose a cluster tree-based method to create virtual SAN that enable both inter-virtual SANs and intra-virtual SAN communications. Virtualization of mobile SANs is also addressed in [4], but in the context of cyber-physical systems. Space and virtual task deadline concepts are introduced in order to address SAN mobility related challenges. The authors also discuss and propose migration algorithms that allow to move virtual resources from one physical resource to another, thus enabling migration of virtual mobile SANs.

1) Research Challenges: Although VNE has been an active area of research for several years now, there are still major challenges related to CARS that need to be addressed. Here are some of these challenges:

a) Enabling embedding decentralization: Distributed algorithms to realize virtual resource embedding need to be investigated while taking into account for factors like ensuring resiliency and robustness to failure, allowing smooth migration of virtual networks from a set of substrates to another, and considering multiple objectives (e.g., security, energy, and QoS) altogether.

b) Addressing environment heterogeneity: Dealing with network links with various types poses new challenges. Embedding virtual links on wireless links or migrating virtual links from wired links to wireless links can be very challenging. Challenges come from multiple factors, all pertaining to the wireless nature, such as mobility and lack of centralization.

c) Increasing sensor availability and reliability: It is indispensable, but also very risky to involve participatory nodes (e.g., smart phones and smart cameras owned by private people) to be part of the distributed sensing platforms and to consider them part of the physical nodes on which virtual networks are to be mapped to. Ensuring sensor availability is a very challenging task which requires development of good pricing schemes that can encourage the participation of sensory nodes. It also requires development of availability and reliability models that take into account probabilistic presence of these nodes.

d) Providing good incentive mechanisms: CARS relies heavily on distributed sensors to collect data. Although most CARS vendors will probably be deploying and relying on their own sensors, vendors will still need to rely on participatory and private sensors to collect their data, which can be essential for enabling distributed sensory platforms. Appropriate pricing schemes will be needed to enable distributed sensing platforms by providing good incentives for these sensors to be part of CARS systems.

E. Intercloud Interoperability and Portability

Enabling CARS with cloud-to-cloud communication and intercloud interoperability capabilities is critical to the success and survival of CARS. As of the time of the writing of this article, twenty one companies and research institutes from over the world have joined the IEEE Intercloud Testbed, an IEEE-led project whose aim is to develop cloud-to-cloud interoperability and federation capabilities, and create a diverse cloud ecosystem that enables cloud services to become ubiquitous.[4] Enabling intercloud interoperability allows data exchange and resource sharing among clouds within the same layer (intra-layer cloud interaction), as well as across clouds belonging to different layers (inter-layer cloud interaction). Inter-layer cloud interactions

[4][Online]. Available: http://cloudcomputing.ieee.org/intercloud/press-release/

facilitate and enable the establishment of inter-layer resource sharing agreements. Intra-layer cloud interactions enable accommodation of unexpected increases in demands and migration of tasks and applications to enhance resource utilization and increase profitability. Clouds are not, however, expected to give up ownership, and they need to preserve independent management and governance of their clouds. In addition to data exchange and resource sharing, there are several other reasons for why enabling intercloud interoperability and portability is important. First, it eases data deployment and migration across different clouds by making it easy to move consumer data in and out of the cloud. Second, it provides customers with the flexibility of selecting, mixing, and/or changing cloud service providers with minimal input and intervention. Third, it adds elasticity to service outsourcing by allowing customers and enterprises to easily move to and from the cloud only some parts of their applications and services while hosting other parts locally. This also allows customers to spread their outsourced applications and software across multiple cloud providers, not forcing them to necessarily use just one provider. Finally, it facilitates adoption of new elements to the clouds and makes it possible for various different platforms, protocols, or technologies to use each other's functionalities. This also eases maintenance and repairs of clouds by allowing cloud shutdowns. Ideally, all of the above should be done with minimal changes to existing software, minimal interaction with cloud providers, and minimal customer input.

1) Research Challenges: Although intercloud interoperability has already been recognized as an important topic that needs investigation, very little research has been done so far to address its challenges. Some key challenges are:

a) Intercloud migration performance metrics: Definitions and derivations of the metrics and thresholds to be used to quantify whether CARS service providers have met their obligated SLAs and whether migration should be triggered still need further investigation.

b) Assessment algorithms of cloud services performance: In order to enable and promote intercloud migration and portability, there must be protocols and tools to be used to assess the cloud services performance metrics. This will allow customers to assess how well a given provider performs and whether to stick around or not.

c) Optimization algorithms of intercloud migration process: Protocols and tools to be used to select the best CARS provider when migration is triggered, as well as policies and regulations to be adopted during this process need to be developed. This will allow customers that choose to migrate to select the best next cloud provider.

d) Standards and ontologies: Different CARS service providers may offer different types of services, may have different ways to describe their services, and may use different types of architectures. Therefore, there is a need for standards and unified ontology and APIs via which services can be specified and requested, and negotiations between CARS providers and their customers can be enabled. Also, data representation, storage, and uniform addressing in clouds need to be standardized to ease interaction and data exchange across clouds. Very little efforts have been put to establish unified APIs and standards for cloud computing in general and for CARS services in particular.

VII. Conclusion

This paper surveys CARS applications and services. The survey starts off by describing the potentials and capabilities of remote sensing when empowered via cloud services. It supports these CARS' capabilities and benefits through applications taken from real-world scenarios. The survey then presents a four-layer architecture for CARS by describing the functionalities, responsibilities, and inter-layer interactions of each layer. The survey then describes three different CARS services models and presents some existing commercial platforms. Finally, it describes key design components that are required for enabling CARS and discusses some of its major challenges.

References

[1] J. Bradley, J. Barbier, and D. Handler, "Embracing the Internet of Everything to capture your share of $14. 4 trillion," 2013.

[2] J. A. Stankovic, "Research directions for the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 3–9, Mar. 2014.

[3] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Sensing as a service model for smart cities supported by Internet of Things," *Trans. Emerging Telecommun. Technol.*, vol. 25, no. 1, pp. 81–93, 2013.

[4] J. Huang, C. M. Kirsch, and R. Sengupta, "Virtual vehicle and cyber-physical cloud computing," 2013.

[5] Z. Chong *et al.*, "Autonomy for mobility on demand," in *Intelligent Autonomous Systems 12*. New York, NY, USA: Springer, 2013, pp. 671–682.

[6] A. Zanella, N. Bui, A. P. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[7] X. Sheng, J. Tang, X. Xiao, and G. Xue, "Sensing as a service: Challenges, solutions and future directions," Cyber-Phys. Cloud Comput. Lab, Univ. California at Berkeley, Berkeley, CA, USA, Tech. Rep., Working Papers CPCC-WP-2013-01-01, 2013, unpublished.

[8] N. D. Lane *et al.*, "A survey of mobile phone sensing," *IEEE Commun. Mag.*, vol. 48, no. 9, pp. 140–150, Sep. 2010.

[9] S. Ehsan *et al.*, "Design and analysis of delay-tolerant sensor networks for monitoring and tracking free-roaming animals," *IEEE Trans. Wireless Commun.*, vol. 11, no. 3, pp. 1220–1227, Mar. 2012.

[10] S. Bhattacharya, S. Sridevi, and R. Pitchiah, "Indoor air quality monitoring using wireless sensor network," in *Proc. 6th Int. Conf. Sens. Technol. (ICST)*, 2012, pp. 422–427.

[11] A. K. Jain, A. Khare, and K. K. Pandey, "Developing an efficient framework for real-time monitoring of forest fire using wireless sensor network," in *Proc. 2nd IEEE Int. Conf. Parallel Distrib. Grid Comput. (PDGC)*, 2012, pp. 811–815.

[12] T. Vaidya, P. Swami, S. Rindhe, S. Kulkarni, and S. Patil, "Avalanche monitoring & early alert system using wireless sensor network," *Int. J. Adv. Res. Comput. Sci. Electron. Eng.*, vol. 2, no. 1, p. 38, 2013.

[13] S. Yadav, K. Swarnkar, T. Chauhan, and R. Jain, "How to improve food security and farming systems by using wireless sensor networks (WSN)," *J. Inf. Eng. Appl.*, vol. 2, no. 6, pp. 11–20, 2012.

[14] B. Hariharan, A. Sasidharan, and A. V. Vidyapeetham, "iWEDS intelligent explosive detection and terrorist tracking system using wireless sensor network," *Int. J. Comput. Sci. Issues*, vol. 8, no. 4, pp. 245–250, 2011.

[15] S. Simi and M. V. Ramesh, "Real-time monitoring of explosives using wireless sensor networks," in *Proc. 1st Amrita ACM-W Celebration Women Comput. India*, 2010, p. 44.

[16] D. Diamond, S. Coyle, S. Scarmagnani, and J. Hayes, "Wireless sensor networks and chemo-/biosensing," *Chem. Rev.*, vol. 108, no. 2, pp. 652–679, 2008.

[17] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2688–2710, 2010.

[18] I. F. Akyildiz, T. Melodia, and K. R. Chowdury, "Wireless multimedia sensor networks: A survey," *IEEE Wireless Commun.*, vol. 14, no. 6, pp. 32–39, Dec. 2007.

[19] N.-Q. Nhan, M.-T. Vo, T.-D. Nguyen, and H.-T. Huynh, “Improving the performance of mobile data collecting systems for electricity meter reading using wireless sensor network,” in *Proc. Int. Conf. Adv. Technol. Commun. (ATC)*, pp. 241–246, 2012.

[20] W. Wilson and G. Atkinson, “Wireless sensing opportunities for aerospace applications,” NASA Langley Res. Center, Hampton, VA, USA, NASA Tech. Rep. LF99-5739, Jan. 2007.

[21] S. Longhi *et al.*, “Solid waste management architecture using wireless sensor network technology,” in *Proc. 5th Int. Conf. New Technol. Mobility Secur. (NTMS)*, 2012, pp. 1–5.

[22] M. Idris, E. Tamil, N. Noor, Z. Razak, and K. Fong, “Parking guidance system utilizing wireless sensor network and ultrasonic sensor,” *Inf. Technol. J.*, vol. 8, no. 2, pp. 138–146, 2009.

[23] P. Mohan, V. N. Padmanabhan, and R. Ramjee, “Nericell: Rich monitoring of road and traffic conditions using mobile smartphones,” in *Proc. 6th ACM Conf. Embedded Netw. Sens. Syst.*, 2008, pp. 323–336.

[24] R. K. Megalingam, V. Mohan, P. Leons, R. Shooja, and M. Ajay, “Smart traffic controller using wireless sensor network for dynamic traffic routing and over speed detection,” in *Proc. IEEE Global Humanit. Technol. Conf. (GHTC)*, 2011, pp. 528–533.

[25] M. R. Raut, “Intelligent public transport prediction system using wireless sensor network,” *Int. J. Comput. Sci. Appl.*, vol. 6, no. 2, 2013.

[26] M. Salathé *et al.*, “A high-resolution human contact network for infectious disease transmission,” *Proc. Nat. Acad. Sci.*, vol. 107, no. 51, pp. 22 020–22 025, 2010.

[27] A. Mpitziopoulos, C. Konstantopoulos, D. Gavalas, and G. Pantziou, “A pervasive assistive environment for visually impaired people using wireless sensor network infrastructure,” *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 194–206, 2011.

[28] Y. Sun, “Human daily activity detect system optimization method using Bayesian network based on wireless sensor network,” in *Advances in Computer Science, Intelligent System and Environment*. New York, NY, USA: Springer, 2011, pp. 721–725.

[29] K. Lorincz *et al.*, “Mercury: A wearable sensor network platform for high-fidelity motion analysis,” in *Proc. 7th ACM Conf. Embedded Netw. Sens. Syst. (SenSys)*, 2009, vol. 9, pp. 183–196.

[30] A. Reehorst *et al.*, “Progress towards the remote sensing of aircraft icing hazards,” *Proc. SPIE*, vol. 7088, pp. 70880J–1, 2008.

[31] B. Hull *et al.*, “Cartel: A distributed mobile sensor computing system,” in *Proc. 4th Int. Conf. Embedded Netw. Sens. Syst.*, 2006, pp. 125–138.

[32] R. Khanna, D. Choudhury, P. Y. Chiang, H. Liu, and L. Xia, “Innovative approach to server performance and power monitoring in data centers using wireless sensors,” in *Proc. IEEE Radio Wireless Symp. (RWS)*, 2012, pp. 99–102.

[33] R. Khanna, H. Liu, and H.-H. Chen, “Self-organization of wireless sensor network for autonomous control in an IT server platform,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2010, pp. 1–5.

[34] S. Choochaisri, V. Niennattrakul, S. Jenjaturong, C. Intanagonwiwat, and C. A. Ratanamahatana, “SENVM: Server environment monitoring and controlling system for a small data center using wireless sensor network,” arXiv preprint arXiv:1105. 6160, 2011.

[35] R. Antonini, G. Fici, and M. Gaspardone, “Energy management of telecommunication plants using wireless sensor network,” in *Proc. 14th Int. Conf. Intell. Next Gener. Netw. (ICIN)*, 2010, pp. 1–6.

[36] J. Yang, S. Phelan, P. Meehan, and S. Daniels, “A distributed real-time sensor network for enhancing energy efficiency through ICT,” in *Proc. Symp. ICT Energy Effic. Workshop Inf. Theory Secur. (CIICT’12)*, 2012, pp. 8–14.

[37] S. Kim *et al.*, “Health monitoring of civil infrastructures using wireless sensor networks,” in *Proc. 6th Int. Symp. Inf. Process. Sens. Netw. (IPSN’07)*, 2007, pp. 254–263.

[38] S. N. Pakzad and G. L. Fenves, “Statistical analysis of vibration modes of a suspension bridge using spatially dense wireless sensor network,” *J. Struct. Eng.*, vol. 135, no. 7, pp. 863–872, 2009.

[39] Q. Huang, R. F. Cox, M. Shaurette, and J. Wang, “Intelligent building hazard detection using wireless sensor network and machine learning techniques,” in *Proc. Comput. Civil Eng.*, 2012, pp. 485–492.

[40] N.-H. Nguyen, Q.-T. Tran, J.-M. Leger, and T.-P. Vuong, “A real-time control using wireless sensor network for intelligent energy management system in buildings,” in *Proc. IEEE Workshop Environ. Energy Struct. Monit. Syst. (EESMS)*, 2010, pp. 87–92.

[41] R. Sharma and K. Sharma, “An adhoc sensor network with an infrared imaging system for controlling room air temperature in a large auditorium building,” *Int. J. Res. Comput. Commun. Technol.*, vol. 1, no. 4, pp. 151–154, 2012.

[42] V. C. Gungor, B. Lu, and G. P. Hancke, “Opportunities and challenges of wireless sensor networks in smart grid,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 10, pp. 3557–3564, Oct. 2010.

[43] A. Sabata and S. Brossia, “Remote monitoring of pipelines using wireless sensor network,” U.S. Patent 7,526,944, May 5, 2009.

[44] F. Alsaade, N. Zaman, A. Abdullah, and M. Z. Dawood, “Enhancing surveillance and security of oil pipelines transportation using wireless sensor network,” *Middle-East J. Sci. Res.*, vol. 11, no. 8, pp. 1029–1035, 2012.

[45] C. Doukas and I. Maglogiannis, “Managing wearable sensor data through cloud computing,” in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, 2011, pp. 440–445.

[46] *Carriots* [Online]. Available: http://www.carriots.com/

[47] *Coversant* [Online]. Available: http://www.coversant.net/

[48] (2013). *Etherios device cloud* [Online]. Available: http://www.etherios.com/

[49] *Ilstechnology* [Online]. Available: http://www.ilstechnology.com/

[50] *Layer7 technologies* [Online]. Available: http://www.layer7tech.com/

[51] *Numerex* [Online]. Available: http://www.numerex.com/

[52] *Arm iot* [Online]. Available: http://www.arm.com/markets/embedded/internet-of-things-iot.php

[53] *ubidots* [Online]. Available: http://www.ubidots.com/

[54] *Livingplanit* [Online]. Available: http://www.living-planit.com/

[55] *arkessa* [Online]. Available: http://www.arkessa.com/

[56] *Thingworx* [Online]. Available: http://www.thingworx.com/

[57] *Axeda* [Online]. Available: http://www.axeda.com/

[58] *Maingate* [Online]. Available: http://www.maingatesolutions. com/

[59] O. Haque *et al.* (2004). *Pachube* [Online]. Available: http://www.pachube.com

[60] *Nimbits* [Online]. Available: http://www.nimbits.com/

[61] (2013). *Thingspeak* [Online]. Available: https://www.thingspeak.com/

[62] A. Pintus, D. Carboni, and A. Piras, “Paraimpu: A platform for a social web of things,” in *Proc. 21st Int. Conf. Companion World Wide Web*, 2012, pp. 401–404.

[63] (2013). *Sensorcloud* [Online]. Available: www.sensorcloud.com/

[64] *Oracle Internet of Things* [Online]. Available: http://www.oracle.com/us/solutions/internetofthings/index.html

[65] Y. Wang, S. Chen, and M. Pedram, “Service level agreement-based joint application environment assignment and resource allocation in cloud computing systems,” in *Proc. IEEE Green Technol. Conf.*, 2013, pp. 167–174.

[66] H. Goudarzi and M. Pedram, “Multi-dimensional SLA-based resource allocation for multi-tier cloud computing systems,” in *Proc. IEEE Int. Conf. Cloud Comput. (CLOUD)*, 2011, pp. 324–331.

[67] Y. Wang, S. Chen, H. Goudarzi, and M. Pedram, “Resource allocation and consolidation in a multi-core server cluster using a Markov decision process model,” in *Proc. 14th Int. Symp. Qual. Electron. Des. (ISQED)*, 2013, pp. 635–642.

[68] Y. Wang, X. Lir, and M. Pedram, “A sequential game perspective and optimization of the smart grid with distributed data centers,” in *Proc. IEEE PES Innovative Smart Grid Technol. (ISGT)*, 2013, pp. 1–6.

[69] G. Wei, A. V. Vasilakos, Y. Zheng, and N. Xiong, “A game-theoretic method of fair resource allocation for cloud computing services,” *J. Supercomput.*, vol. 54, no. 2, pp. 252–269, 2010.

[70] K. H. Kim, A. Beloglazov, and R. Buyya, “Power-aware provisioning of cloud resources for real-time services,” in *Proc. 7th Int. Workshop Middleware Grids Clouds e-Sci.*, 2009, p. 1.

[71] N. M. K. Chowdhury and R. Boutaba, “Network virtualization: State of the art and research challenges,” *IEEE Commun. Mag.*, vol. 47, no. 7, pp. 20–26, Jul. 2009.

[72] A. Fischer, J. Botero, M. Beck, H. De Meer, and X. Hesselbach, “Virtual network embedding: A survey,” *IEEE Commun. Surveys Tuts.*, no. 99, pp. 1–19, 2013.

[73] N. McKeown *et al.*, “Openflow: Enabling innovation in campus networks,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.

[74] T. Taleb and A. Kunz, “Machine type communications in 3gpp networks: Potential, challenges, and solutions,” *IEEE Comm. Mag.*, vol. 50, no. 3, pp. 178–184, Mar. 2012.

[75] J. Huang, C. Krainer, C. M. Kirsch, and R. Sengupta, “Service provisioning in cyber-physical cloud computing,” Cyber-Phys. Cloud Comput. Lab, Univ. California at Berkeley, Berkeley, CA, USA, Tech. Rep., Working Paper CPCC-WP-2012-11-01, 2012, unpublished.

[76] P. Saripalli, G. Kiran, R. R. Shankar, H. Narware, and N. Bindal, “Load prediction and hot spot detection models for autonomic cloud computing,” in *Proc. IEEE Int. Conf. Util. Cloud Comput. (UCC)*, 2011, pp. 397–402.

[77] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *J. Internet Serv. Appl.*, vol. 1, no. 1, pp. 7–18, 2010.

[78] C. Dabrowski and F. Hunt, "Using Markov chain analysis to study dynamic behaviour in large-scale grid systems," in *Proc. ACM Symp. Grid Comput. e-Res.*, 2009, pp. 29–40.

[79] Y. Zhang, W. Sun, and Y. Inoguchi, "CPU load predictions on the computational grid," in *Proc. IEEE Int. Symp. Cluster Comput. Grid (CCGRID)*, 2006, pp. 40–47.

[80] Y. Wu, Y. Yuan, G. Yang, and W. Zheng, "Load prediction using hybrid model for computational grid," in *Proc. ACM Int. Conf. Grid Comput.*, 2007, pp. 235–242.

[81] T. Heath, B. Diniz, E. Carrera, W. Meira, Jr., and R. Bianchini, "Energy conservation in heterogeneous server clusters," in *Proc. ACM SIGPLAN Symp. Princ. Pract. Parallel Programm.*, 2005, pp. 186–195.

[82] J. Prevost, K. Nagothu, B. Kelley, and M. Jamshidi, "Prediction of cloud data center networks loads using stochastic and neural models," in *Proc. IEEE Int. Conf. Syst. Syst. Eng. (SoSE)*, 2011, pp. 276–281.

[83] V. Kantere, D. Dash, G. Francois, S. Kyriakopoulou, and A. Ailamaki, "Optimal service pricing for a cloud cache," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 9, pp. 1345–1358, Sep. 2011.

[84] C. S. Yeo, S. Venugopal, X. Chu, and R. Buyya, "Autonomic metered pricing for a utility computing service," *Future Gener. Comput. Syst.*, vol. 26, no. 8, pp. 1368–1380, 2010.

[85] C. Perera *et al.*, "Sensor search techniques for sensing as a service architecture for the Internet of Things," *IEEE Sensors J.*, vol. 14, no. 2, pp. 406–420, Feb. 2014.

[86] T. J. Trappler, "If it's in the cloud, get it on paper: Cloud computing contract issues," *Educause Quart.*, vol. 33, no. 2, p. n2, 2010.

[87] J. E. Pezoa and M. M. Hayat, "Reliability of heterogeneous distributed computing systems in the presence of correlated failures," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 4, pp. 1034–1043, Apr. 2014.

[88] A. Shrestha, L. Xing, and H. Liu, "Modeling and evaluating the reliability of wireless sensor networks," in *Proc. Annu. Rel. Maintain. Symp. (RAMS'07)*, 2007, pp. 186–191.

[89] A. Ghosh and S. K. Das, "Coverage and connectivity issues in wireless sensor networks: A survey," *Pervasive Mobile Comput.*, vol. 4, no. 3, pp. 303–334, 2008.

[90] S. K. Garg, S. K. Gopalaiyengar, and R. Buyya, "Sla-based resource provisioning for heterogeneous workloads in a virtualized cloud datacenter," in *Algorithms and Architectures for Parallel Processing*. New York, NY, USA: Springer, 2011, pp. 371–384.

[91] K. Nahrstedt, "(latest version) SCDA: Sla-aware cloud datacenter architecture for efficient content storage and retrieval," Tech. Rep., 2013, unpublished.

[92] S. Alam, M. M. Chowdhury, and J. Noll, "Senaas: An event-driven sensor virtualization approach for Internet of Things cloud," in *Proc. IEEE Int. Conf. Netw. Embedded Syst. Enterp. Appl. (NESEA)*, 2010, pp. 1–6.

[93] I. Khan, F. Belqasmi, R. Glitho, and N. Crespi, "A multi-layer architecture for wireless sensor network virtualization," in *Proc. 6th Joint IFIP Wireless Mobile Netw. Conf. (WMNC)*, 2013, pp. 1–4.

[94] S. Distefano *et al.*, "A hypervisor for infrastructure-enabled sensing clouds," *IEEE Int Conf. Commun. Workshops (ICC)*, 2013, pp. 1362–1366.

[95] H. Bandara, A. P. Jayasumana, and T. H. Illangasekare, "Cluster tree based self organization of virtual sensor networks," in *Proc. IEEE GLOBECOM Workshops*, 2008, pp. 1–6.

Sherif Abdelwahab (S'07–M'11) received the B.S. and M.S. degrees in electronics and communications engineering from Cairo University, Giza, Egypt, in 2004 and 2010, respectively, and is currently working toward the Ph.D. degree at the School of Electrical Engineering and Computer Science (EECS), Oregon State University, Corvallis, OR, USA.

Before pursuing the Ph.D. degree, he was with the mobile networks industry with Alcatel-Lucent from 2004 to 2007 and with Etisalat from 2008 to 2013. His research interests include design, analysis, and optimization of networked systems enabling Internet of Things.

Bechir Hamdaoui (S'02–M'05–SM'12) received the Diploma of Graduate Engineering from the National School of Engineers, Tunis, Tunisia, in 1997, and the M.S. degrees in electrical and computer engineering (ECE) and computer science (CS) and the Ph.D. degree in computer engineering from the University of Wisconsin–Madison, Madison, WI, USA, in 2002, 2004, and 2005, respectively.

Currently, he is an Associate Professor with the School of Electrical Engineering and Computer Science (EECS), Oregon State University, Corvallis, OR, USA. He also served as an Associate Editor for the *Journal of Computer Systems, Networks, and Communications* from 2007 to 2009. His research interests include distributed resource management and optimization, parallel computing, cooperative and cognitive networking, cloud computing, and the Internet of Things.

Dr. Hamdaoui is presently an Associate Editor for the IEEE Transactions on Vehicular Technology from 2009 to present, the IEEE Transactions on Wireless Communications from 2013 to present, and *Wireless Communications and Computing Journal* from 2009 to present. He served as the Chair for the 2011 ACM MobiCom SRC program, and as the Program Chair/Cochair of several IEEE symposia and workshops (including ICC 2014, IWCMC 2009–2014, CTS 2012, and PERCOM 2009). He also served on Technical Program Committees of many IEEE/ACM conferences, including INFOCOM, ICC, GLOBECOM, and PIMRC. He was the recipient of the NSF CAREER Award in 2009.

Mohsen Guizani (S'85–M'89–SM'99–F'09) received the B.S. (with distinction) and M.S. degrees in electrical engineering and the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively.

He is currently a Professor and Associate Vice-President of Graduate Studies with Qatar University, Qatar. Previously, he served as the Chair with the Computer Science Department, Western Michigan University, Kalamazoo, MI, USA, from 2002 to 2006, and Chair of the Computer Science Department, University of West Florida, Pensacola, FL, USA, from 1999 to 2002. He also served in academic positions with the University of Missouri–Kansas City, Kansas City, MO, USA; University of Colorado–Boulder, Boulder, CO, USA; Syracuse University; and Kuwait University, Kuwait. He currently serves on the editorial boards of many international technical journals and the Founder and EIC of Wireless Communications and Mobile Computing Journal published by John Wiley (available: http://www.interscience.wiley.com/jpages/1530-8669/). He is the author of 9 books and more than 300 publications in refereed journals and conferences. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing and smart grid.

Dr. Guizani guest edited a number of special issues in IEEE journals and magazines. He is also the Founder and General Chair of the International Conference of Wireless Communications, Networking and Mobile Computing (IWCMC). He also served as a member, Chair, and General Chair of a number of conferences. He was selected as the Best Teaching Assistant for two consecutive years from Syracuse University, 1988 and 1989, respectively. He was the Chair of the IEEE Communications Society Wireless Technical Committee and Chair of the TAOS Technical Committees. He served as the IEEE Computer Society Distinguished Speaker from 2003 to 2005. He is a member of ASEE and a Senior Member of the ACM.

Ammar Rayes received the B.S. and M.S. degrees from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, and the D.Sc. degree both in electrical engineering from Washington University, St. Louis, MO, USA.

He is a Distinguished Engineer with Cisco Systems and the Founding President with The International Society of Service Innovation Professionals (available: http://www.issip.org). He is currently chairing Cisco Services Research Program. He has authored/coauthored over a 100 papers and patents on advances in communications-related technologies, including a book on network modeling and simulation and another one on ATM switching and network design. He is an Editor-in-Chief for *Advances of Internet of Things Journal* and served as an Associate Editor of the *ACM Transactions on Internet Technology* and on the *Journal of Wireless Communications and Mobile Computing*. His research interests include smart services, Internet of Everything (IoE), machine-to-machine, smart analytics, and ip strategy.

Dr. Rayes was the recipient of the Outstanding Graduate Student Award in telecommunications from Washington University.