IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 28, 2019, accepted October 7, 2019, date of publication October 17, 2019, date of current version November 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2948027

Service Management for IoT: Requirements, Taxonomy, Recent Advances and Open Research Challenges

ABDELMUTTLIB IBRAHIM ABDALLA AHMED[1], (Member, IEEE), ABDULLAH GANI[1,2], (Senior Member, IEEE), SITI HAFIZAH AB HAMID[1,3], ABDELZAHIR ABDELMABOUD[4], HASSAN JAMIL SYED[5], RIYAZ AHAMED ARIYALURAN HABEEB MOHAMED[1,6], AND IHSAN ALI[1]

[1]Center for Mobile Cloud Computing Research, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia
[2]Faculty of Computing and Informatics, Universiti Malaysia Sabah, Kota Kinabalu 88400, Malaysia
[3]Department of Software Engineering, University of Technology Malaysia, Johor Bahru 81310, Malaysia
[4]Department of Information Systems, King Khalid University, Muhayil Asir 61421, Saudi Arabia
[5]Department of Computer Science, National University of Computer and Emerging Sciences, Karachi 75030, Pakistan
[6]FSTEM, International University of Malaya-Wales, Kuala Lumpur 50480, Malaysia

Corresponding authors: Abdelmuttlib Ibrahim Abdalla Ahmed (abdelmuttlib@siswa.um.edu.my), Abdullah Gani (abdullah@um.edu.my), and Siti Hafizah Ab Hamid (sitihafizah@um.edu.my)

This work was supported by the Deanship of Scientific Research at King Khalid University under Grant R.G.P-319-2019.

ABSTRACT The exponential growth in the number of smart devices connected to the Internet of Things (IoT), and associated with various IoT-based smart applications and services, raises interoperability challenges which could affect the sustainability of IoT services. IoT software applications are built using different software platforms and embedded in diverse types of terminals and sensing devices. Aiming to offer smart services over a range of network technologies that use different communication protocols. The concept of Web service with service-oriented solutions was introduced to cope with the heterogeneity of hardware and software, and to tackle issues of interoperability, flexibility and scalability. The main step of this solution was the integration of Web of Things technologies into smart device networks, with the utilization of IoT gateways. Service management is a crucial factor in sustaining service-oriented solutions in dynamic and highly scalable IoT systems, and is concerned with several issues associated with service provisioning, orchestration, composition and adaption. This work was motivated by the need for robust and flexible service management systems that can meet the requirements for the rapid scalability and heterogeneity associated with the exponential growth of IoT systems. In the literature there is no survey of service management issues and associated research efforts in the field of IoT. In this article, we identify the key requirements for managing IoT services as well as common service management platforms for IoT. We provide a thematic taxonomy based on the important factors, and investigate recent advances in service management for IoT systems. Finally, the major challenges that remain open are presented as a guide for future research directions.

INDEX TERMS Internet of Things, web service, web of things, SOA, microservice, service composition, service orchestration.

I. INTRODUCTION

Rapid advancements in emerging technologies and the smooth convergence of wireless communication, sensors and radio frequency identification (RFID) have resulted in the birth of the Internet of Things (IoT). IoT service platforms and corresponding smart features have been embedded in electromechanical systems and controllers to establish seamless integration between the physical world and cyberspace and to provide smart service via daily life applications [5].

Numerous IoT platforms and connectivity protocols have been developed, for instance the constrained application protocol (CoAP), Bluetooth low energy (BLE) and message queuing telemetry transport (MQTT). However, the heterogeneity of the IoT devices, standards and communication protocols raises several problems, such as a lack of

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

interoperability, scalability and flexibility [7]. A service orientation is utilized in many studies to discuss these problems.

Service-oriented solutions include service-oriented architecture (SOA) and microservice architecture, which are architectural patterns followed in IoT design [7], [10]. In a service-oriented IoT, the devices/entities provide services to other devices/entities via communication protocols. A service is ''a discrete unit of business functionality that is made available through a service contract'' [18]. A service contract comprises a service interface, documentation, QoS and service policies. The term 'service management' indicates a method of enabling seamless service composition, integration and interoperability among various IoT applications and platforms, which run on various devices over heterogeneous networking technologies [7]. Service management aims to ensure and monitor the performance and quality of service of IoT transactions.

Service-oriented solutions integrate web services into sensor networks via the utilization of IoT-optimised gateways that can fill the gap between devices, networks and access terminals [14].

The World Wide Web Consortium (W3C) introduced web service through Web of Things (WoT) technologies in order to ensure interoperability and integration between IoT platforms and domains by extending web technologies (e.g. metadata and APIs). WoT provides a Thing description (TD) mechanism that is used to describe IoT interfaces, and via which IoT technologies and services communicate with each other regardless of the underlying details and heterogeneity, across multiple networking protocols [15]. In addition, WoT provides a standardized approach for defining and programming IoT behaviors.

In service-oriented solutions, there are three groups of actors: a set of service providers, a set of service requestors and directories of Things. A WoT server represents the service providers (by publishing their services at runtime), and a WoT client represents service requestors (by discovering the published services). WoT servers, clients and device control methods are contained in an entity called a WoT servient. The service requester selects a service provider based on the type and quality of the offered service [16]. Figure 1 illustrates the processes in a service-oriented solution adapted to IoT.

This article is the first comprehensive survey of service management for IoT, although service composition has previously been reviewed as one the factors of service management [17]. This study is conducted with the aim of investigating service management for IoT, and is motivated by the need for service-oriented solutions, as improved versions will be needed in the future to accommodate a tremendous number of services offered by devices through heterogeneous communication networks and protocols.

The contributions of this article are summarized as follows:

- We identify and describe the core requirements of service management in IoT.
- We extensively investigate service management in the context of IoT, and articulate some prominent recent advances.
- We investigate service management platforms in IoT to identify service management paradigms.
- We present a thematic taxonomy based on the most important parameters as an aid to understanding service management and its associated issues in IoT.
- We include a discussion of service management challenges, as a guide for future research.

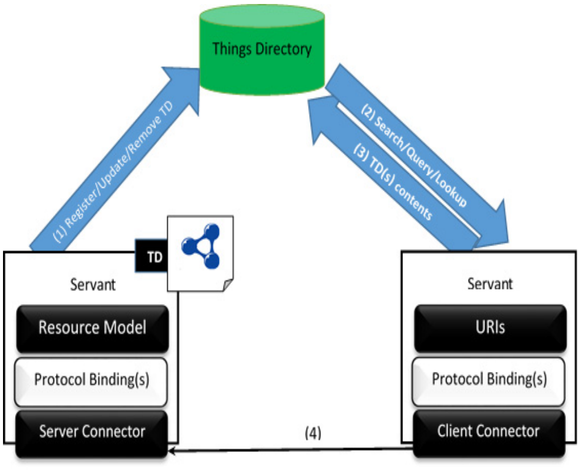

FIGURE 1. Service-oriented solution in the IoT.

Each of these contributions is presented in a separate section, from Sections II to VII, and the conclusion is then provided in Section VIII.

II. BACKGROUND

This section presents the basic concepts and definitions related to service management in IoT.

A. THE CONCEPT OF SERVICE IN IoT

The services are self-contained, loosely-coupled, platform-independent, discoverable, composable and invokable [19]. 'Self-contained' means that a service maintains its own state independently from the application that uses it, while 'loosely-coupled' means that there are few dependencies between a consumer and a service. 'Platform-independent' means that a requestor can invoke the service regardless of the differences between the platforms (hardware or software).

The most common way of applying the concepts of service is through web service technologies, i.e. computing technologies that enable data exchange and interoperability between different applications running on various devices over the web [20]. WoT services utilize several standards and technologies such as JSON, XML, HTTP, MQTT and CoAP. W3C defined a web service as ''a software application identified by a Uniform Resource Identifier (URI), whose interface and binding are capable of being defined, described and discovered by XML artefacts to support direct interactions with other software applications using XML based messages

TABLE 1. Defintion and terms of WoT.

Term	Definition
Thing	The term 'Thing' is used as an abstract representation of a virtual or physical entity that plays some role in IoT applications. An entity can be either a device or a logical component of the device. Things are exposed to IoT applications as software objects with APIs.
WoT Interface	A WoT interface is an API that aims to support Thing interactions over non-web protocols such as ZigBee or MQTT. WoT interfaces involve formal models of IoT resources.
Scripting API	This is an optional way of generating interfaces provided with application scripts to enable a runtime system for IoT applications, in a similar way to a Web browser. Scripting APIs support the interaction, with high productivity and low integration cost, with Things that have a TD but do not have a WoT interface.
Thing Description	A TD is a data structure designed to contain general metadata about a Thing and metadata regarding its services, data model, communication, and security mechanism. Use of TDs promotes interoperability in two ways: (a) by enabling device-to-device communication through the WoT; and (b) by serving as a common, uniform format allowing developers to document and retrieve all the necessary details.
Formal Interaction Model	A formal model that enables multiple messaging styles (publish-subscribe, request-response, and message passing). The default interaction patterns in the model are property, action, and event, where a property represents readable and writable data points, an action is a process that may run for a particular time, and an event is a pattern of action in which a remote endpoint pushes data asynchronously.
WoT Binding Templates	A collection of metadata blueprints for communication, which explains the method of interaction between different IoT platforms. A binding template is used in the configuration of communication stacks to generate compliance messages for various targets in terms of protocols and platforms.
Servient	A software stack deployed in all IoT devices to enable WoT processes. A servient supports service management in IoT through hosting and exposing a service on a service provider Thing, and facilitating service consumption for a service requestor Thing. A servient comprises multiple binding protocols to ensure interoperability among diverse service platforms.

via Internet-based protocols'' [21]. WoT service management can be defined as a management system for WoT services via methods that enforce monitoring, control and notifications of service specifications and quality. Service quality indicates availability (e.g. the appearance of the service on one or more IoT device or cloud of Things), performance (e.g. delay and failure rates) and accessibility to the service via the dashboard and WoT browser.

B. WEB OF THINGS

WoT was introduced by W3C [22] to leverage web standards and technologies (e.g. metadata and APIs) for interconnecting all types of devices, either directly or via an M2M gateway. WoT enables the exposure of functionalities through RESTful APIs, and this supports easy access and interaction and consequently sustains flexible, scalable and interoperable services. Table 1 presents an explanation of the terms and definitions used in WoT.

III. SERVICE MANAGEMENT PLATFORMS FOR IoT

These platforms are software that can offer integrated services, such as simultaneous connectivity among a tremendous number of IoT devices and easily enabling device configuration for device-to-device communication and synchronization with the IoT cloud. This section investigates the IoT platforms that support service management, and a comparison is provided in Table 2.

A. SALESFORCE THUNDER IoT

Salesforce enriched IoT by introducing the its Thunder platform, an event processing and rules engine [23]. This platform was designed to collect, analyze and respond to massive and scalable events in real time. An analysis of data streaming in IoT environments supports predictive and proactive actions. Thunder supports Salesforce's IoT cloud, which can interconnect the big data generated by IoT devices with the consumer's dashboards, other applications and partners, and can initiate actions for real-time responses [24]. Salesforce's IoT cloud assists the customer in understanding the behavior of products and devices by maintaining device profiles based on the customer's context and the data stream received from the IoT device. Streaming data helps in inferring the context in which the device is used. Context data shape information about the activities by combining the customer's details with device data.

B. AMAZON WEB SERVICE IoT

Amazon introduced an integrated solution for service management for IoT, involving several platforms components such as the Amazon Web Service (AWS) IoT Cloud, AWS IoT device management, AWS IoT Device Defender and AWS IoT Analytics [25]. AWS IoT Cloud is a cloud-based platform that smoothly connects IoT devices, and securely enables them to interact with each other and with cloud applications. This platform supports devices and messages by processing and routing those messages to AWS endpoints and other devices in a reliable and secure way [26]. AWS IoT device management is a service that offers friendly and secure onboard monitoring, organization and remote control of scalable IoT devices. The AWS IoT Device Defender is a security service for protecting IoT devices through conducting continuous audits on security policies of IoT devices to ensure that there is no security violations and misbehaviors.

TABLE 2. Comparison OF IoT platforms.

Platform	Interface type	Data collection protocols	Data analytics	Security support
Salesforce Thunder IoT	REST APIs	MQTT, HTTP	Real-time batch and stream data analytics	Message encryption, authentication token
Amazon Web Service (AWS) IoT	REST APIs	MQTT, HTTP	Real-time data analytics	Link encryption , authentication code
Google cloud IoT	REST APIs	MQTT, HTTP	Real-time batch and stream data processing	Fine-grained identity and access management, link encryption
Microsoft Azure IoT hub	REST APIs	MQTT, HTTPS	Real-time stream data analytics	Link encryption, message authentication code, digital signature
Cisco IoT	REST APIs	MQTT, HTTP	Real-time stream data analytics	Traffic encryption, authentication and safeguarding sensitive information from theft
AT&T IoT platform	REST APIs	MQTT, HTTP	Real-time data analytics and visualization	End-to-end link encryption, device authentication
IBM Watson IoT platform	REST and real-time APIs	MQTT, HTTPS	Real-time data analytics	Link encryption, authentication , identity management
Oracle IoT	REST API	MQTT	Real-time and predictive analytics on streams of big data	Link encryption, authentication
Ubidots IoT	REST API	HTTP, MQTT and TCP/UDP	Real-time data visualization and analytics	Token-based authentication, link encryption

AWS IoT Analytics provides accurate and advanced analytics services for massive volumes of IoT data.

C. GOOGLE CLOUD IoT

Google Cloud IoT platform [27] enriches service management for IoT by providing secure connection, data processing and management for millions of globally deployed sensing entities. Cloud IoT, in collaboration with other services on the Cloud IoT platform, offers integrated solutions. These solutions include data aggregation, processing, analysis, and visualization in real time, to ensure operational efficiency. Cloud IoT uses sub-underneath/cloud-pub to aggregate the data from IoT devices into a unified global system that seamlessly integrates the data with data analytics services. Data analytics supports advanced analysis, visualizations and artificial intelligence mechanisms to boost operational efficiency and business optimization. Furthermore, the Google Cloud IoT platform is highly scalable, since it runs on server-less infrastructure and supports standard data transmission and security protocols.

D. MICROSOFT AZURE IoT HUB

Azure IoT Hub is a scalable cloud-IoT platform that comprises a device registry, data storage management, and security services [28]. The platform maintains individual identities and authorizations for each of the connected devices, and preserves the confidentiality of device-to-cloud and cloud-to-device communication. The platform provides a service interface for supporting the development of IoT applications, device synchronization and flexible monitoring. Azure IoT Central and Azure IoT Edge were introduced with Azure IoT Hub to support many operational services. Azure IoT Central facilitates the connection of IoT devices, provides data analytics, and supports businesses integration. Azure IoT Edge realises hybrid cloud and IoT solutions via orchestration between code and services, and between cloud and edge.

E. CISCO IoT

Cisco IoT [29] introduced a mobility-cloud-based software suite for IoT to improve IoT-based businesses. This platform provides numerous services, such as connectivity services, operation management, data management and security. Cisco industrial networking solutions offer reliable and secure connectivity for IoT systems. The Cisco IoT platform manages and runs IoT operations smoothly and consistently with the help of tools such as Cisco IOx for controlling edge applications and Cisco DNA for infrastructure integration. Cisco also introduced Kinetic IoT to support data extraction and computation. Cisco protects the deployment of devices via a secure IoT architecture, which enhances all IoT security services.

F. AT&T IoT

The AT&T IoT Platform [30] enables device manufacturers and developers to build elegant solutions for complex problems. The platform categorizes its services for IoT environment as device services, data services, containerization services, and global connectivity and management. Device services involve management and organization of a scalable registry of devices, utilizing a message broker to issue commands to IoT applications and maintaining the device state (storage and retrieval of state information). Data-related services involve data storage, time-series, data analytics and visualization to support decision making. The containerization service provided as a cloud application is built and deployed in multi-datacenter environments. The AT&T IoT platform provides global connectivity through IoT SIM and hardware.

G. IBM WATSON IoT

The IBM Watson IoT platform [31] follows a cloud-based service management paradigm, and supports secure connectivity among IoT components, data management and analysis. This platform provides an add-on to enable a Blockchain service for the validation and authentication of assets and events. The IBM Watson IoT platform provides analytics service for enriching, augmenting, interacting and smoothly integrating the IoT raw data. This platform makes a significant contribution in terms of increasing business revenue, using bidirectional communication with the end user to accelerate the access of new services and products.

H. ORACLE IoT

Oracle IoT [32] supports several different domains, such as the supply chain and enterprise planning. Moreover, the platform extends the application of human experience to the cyber-physical, to generate new types of applications such as auto-driving using intelligent prediction. The platform provides many services such as asset, production and service monitoring. Asset monitoring is performed via a dedicated IoT cloud service application that aggregates data regarding the location, condition and utilization status of the IoT device. Production monitoring is conducted by collecting and integrating data on the manufacturing machines, production line and factory setting. Service monitoring increases visibility, and supports proactive maintenance through the production prediction paradigm.

I. UBIDOTS IoT

The Ubidots solution [33] supports the industry by connecting various projects such as healthcare, utilities, energy systems, manufacturing and smart transportation through providing data captured from IoT devices, data analytics, events and alarms and live dashboards. Data capturing is achieved by connecting IoT devices to the Ubidots cloud and managing data sensing via device libraries. Data analytics services are provided by the Ubidots engine to improve the efficiency and effectiveness of the IoT system. The events and alarm engine sends an alert in the form of an email, SMS or telegram to the owner of the IoT devices. A live dashboard enables the user/administrators to perform live activities such as device control and data analysis.

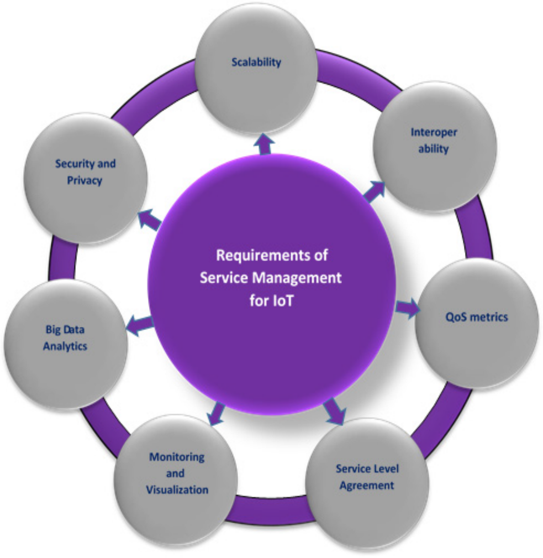

FIGURE 2. Requirements of service management for IoT.

IV. REQUIREMENTS OF SERVICE MANAGEMENT FOR IoT

This section presents the essential requirements of service management for cloud-based IoT systems. Figure 2. represents these requirements.

A. SCALABILITY

In IoT applications, the transactions are generally composed of many services from different service providers. Scalability and flexibility are required, since the number of IoT devices is rapidly increasing and is expected to provide billions of services in the future [34]. Publishing a tremendous amount of resources in the cloud requires a highly scalable directory/registry of Things, in order to ensure the rapid and real-time discovery of IoT resources and services. Unlike many conventional distributed systems, resources in IoT systems are related to each other both semantically and contextually. The traffic monitoring application may invoke a service that involves resources located within the same environmental context, e.g. a monitoring service for a road with light and smoke sensors.

B. INTEROPERABILITY

According to the IEEE [35], interoperability is ‘‘the ability of two or more systems or components to exchange information and to use the information that has been exchanged’’. The essential concern is to allow two or more heterogeneous resources to interact by making their services compatible at the syntactic interface level [36]. A service management platform is required in order to provide interoperable services for the heterogeneous devices, apps, platforms and communication technologies [37]. Interoperability is improved by considering compatibility with communication protocols and standards and all types of applications, such as mobile, business and desktop apps. WoT is trying to tackle this heterogeneity by including communication metadata in the TD.

C. QOS METRICS AND MEASUREMENT

Interactive IoT applications have functional and non-functional requirements for service composition [38]. Functional requirements present the expected functioning service from the IoT service, while the TD defines the functional requirements. Non-functional requirements are QoS issues such as interoperability, response time, availability, accuracy, price reliability, sustainability, and service level agreements (SLAs). Specification of the metrics that determine the service consumer's satisfaction with the provided service is an essential issue for measuring and assessing IoT services. Service providers use QoS metrics to ensure that the service is running according to a specific set of measures, and the service consumer uses specific QoS metrics to select suitable service providers. QoS metrics differ according to the context of use, and have different requirements and degrees of importance [39], [40].

D. MANAGEMENT OF SERVICE LEVEL AGREEMENTS

An SLA is a contract between a service provider and service requestor/consumer. In IoT, web services interact with each other to ensure QoS via SLAs. SLAs are critical for the deployment of IoT entities and the adoption of cloud services [39]. The SLA lifecycle has requirements related to information handling, creation of the SLA template, the definition of management issues and SLA enforcement [41]. Information handling focuses on the processing of the information, which affects the usage of the service. The creation of the SLA template is associated with capturing relevant information such as service components and service provisioning information. Management issues involve several activities such as monitoring, negotiation, assessment, trustworthiness and violation. SLA enforcement and termination are triggered by the time expiration of the service period or a violating action.

E. MONITORING AND VISUALIZATION

IoT services span multiple network domains and sophisticated technologies. A robust service monitoring solution is necessary for critical devices such as alarm systems, IP cameras, smart locks, pet monitors, healthcare devices, thermostats and the cloud of Things, while the automated monitoring and visualization of service provisioning assist in tuning the QoS and scaling network resources to fit the SLA. One of the main challenges facing service management in IoT is the monitoring of SLA violations. IoT allows users to compose massive, pervasive and complicated applications. Consequently, it is crucial to develop an effective method for SLA monitoring and management [39]. Monitoring mechanisms mainly focus on the service provider, and create alerts in the case of bottleneck, failure or SLA violation.

F. BIG DATA ANALYTICS

Recently, big data analytics has been widely utilized in service management processes such as making correlations, deriving deep insights, and extracting patterns from IoT data [42]. These processes help in increasing the operational efficiency and high control in real time. The primary emphasis of big data analytics for service management is the analysis and evaluation of big data records related to QoS and the behaviors of IoT devices. Using real-time big data analytics for IoT performance and activities helps in proactive maintenance and other actions such as storing streaming data in an operational database. Real-time analytics for operational data also optimizes the way in which IoT devices and applications interact, and provides smart services [43].

G. SECURITY AND PRIVACY

Service management for IoT is associated with several security factors; it starts from a consideration of security features in the initial design of the IoT devices and includes the runtime of the service and the method of interaction between these devices [44]. Various security services are required for the different phases of IoT operations, such as device connection and synchronization, preserving the privacy of the transaction data and ensuring the integrity of entities. When an entity connects to the system, authentication is needed to establish trust with other IoT devices and services. After the establishment of trust, IoT devices and services can securely communicate and collaborate via information exchange and performing transactions. Preserving the privacy and integrity of the exchanged data is a crucial issue, as some of the data are sensitive and are used in critical decision making [45]. The incorporation of a reliable lightweight cryptographic mechanism with each IoT device and the application of security practices are required as countermeasures for various security threats.

V. TAXONOMY OF SERVICE MANAGEMENT FOR IoT

This section introduces a thematic taxonomy of service management for IoT that includes several components and parameters of the service management environment, namely service types, architectural organizations, middleware, runtime management, security and applications, as illustrated in Figure 3.

A. TYPES OF SERVICE

IoT services can be categorized based on their technical features into four classes: identity-related, data aggregation, collaborative-aware, and pervasive services.

1) IDENTITY-RELATED SERVICES

Managing the identity of Things and their owners is a crucial factor in successfully leveraging the shifting characteristics of cyberspace. Identity-related services provide an identification feature for IoT entities through several technologies such as RFIDs and barcodes. An RFID tag is attached to a

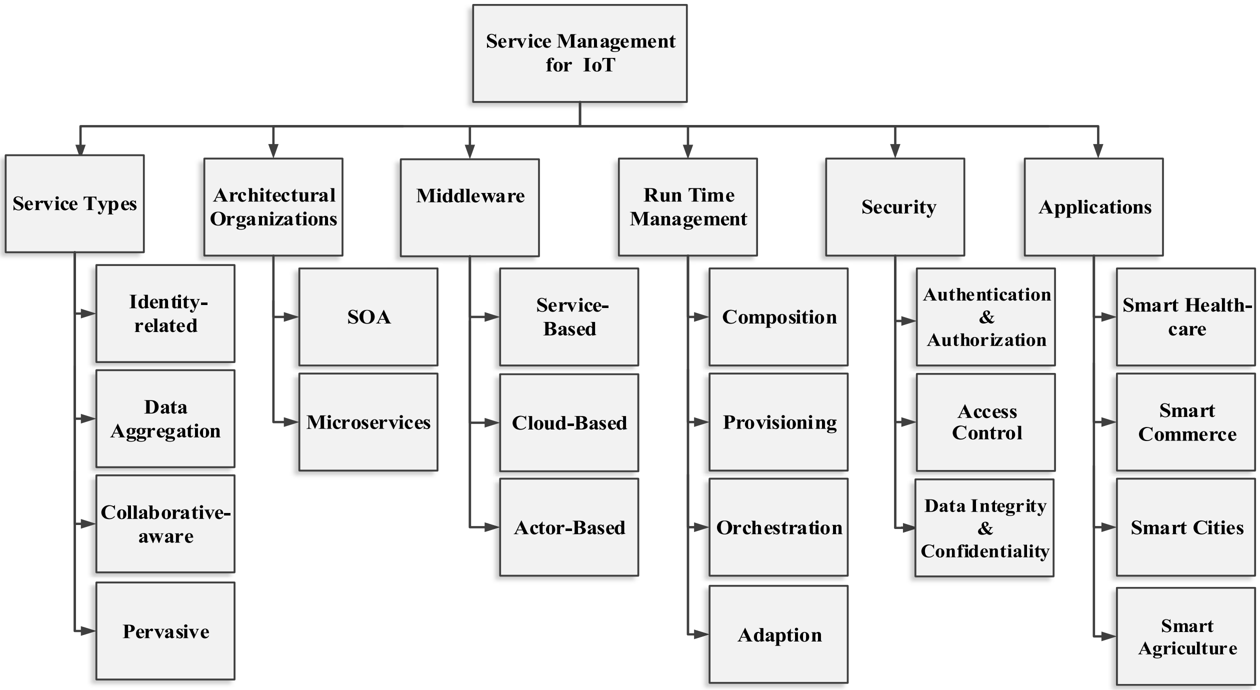

FIGURE 3. Taxonomy of service management for IoT.

device, and a tag reader then accesses the information in the RFID tag (i.e. the identity information), and makes a request to the IoT name resolution server [46].

2) DATA AGGREGATION SERVICES

This type of service utilizes identity-related services in conjunction with the Internet, sensing devices and gateways. IoT gateways enable data aggregation services to access differently to access remote sensors and networking auxiliary devices. Data aggregation is then performed by a sensing device that collects and processes the data and transmits it via a WoT service using JSON or XML to the platform for further processing [47]. The platform implements the management strategy, involving the sensing devices, applications, data, services and third parties. These types of services are necessary in the monitoring of IoT applications, for instance monitoring and control systems in the greenhouse of smart agriculture applications [15].

3) COLLABORATION-AWARE SERVICES

In IoT, a collaboration-aware service involves device-to-device communication and device-to-human interaction, and in many scenarios these types of communication are performed with the help of an IoT cloud [47]. Composing collaboration-aware services requires network security, sensing devices with processing capabilities and smarter terminals. Collaboration-aware services depend on aggregation services to begin assessments for decision making and performing actions.

4) PERVASIVE SERVICES

Pervasive services are the main goal of IoT, and extend collaborative-aware services to provide smooth connectivity anywhere, anytime, for everything, and for every task, whether via computer, smartphone or another type of smart terminal. The RESTful protocol supports pervasive services by providing a universal API that ensure interoperability in IoT systems [42]. To achieve the user's transaction goals, collaborative-aware and pervasive services follow many processes of service management, such as service identifications and definition, service discovery and selection, service composition and service orchestration.

B. ARCHITECTURAL ORGANIZATIONS

Architectural organization is used to model high-level design that fulfils the requirements of various actors when building an IoT application. This architecture further provides direction for application design and development, which contains layers and tiers. The most common architectural organization for IoT services are service-oriented architecture (SOA) and micro-services, which are discussed in the following and compared in Table 3.

1) SERVICE-ORIENTED ARCHITECTURE

SOA for IoT (SOA-IoT) is used to couple heavyweight functionalities or containers in a corporate information system. Moreover, SoA is suitable for embedding many real-world devices to assist in the processing and communication of user tasks [48]. SOA-IoT-based load balancing techniques

TABLE 3. IoT service management architectures.

Factors	MSA-IoT	SOA-IoT
Focus domain	Business functionality	Device capabilities
Self-containment	Fewer dependences, libraries packed with application	Libraries not packed with application
Service deployment	Does not depend on application server	Uses common IoT platforms
Virtualization-based deployment	Docker-based	OSGi-based, Docker-based
Integration	Fully	Partially
Continuous delivery	Yes	No
Update	Short release cycles	Rare updates
Communication approach	Simple messaging system	IoT Enterprise service bus
Protocols	HTTP,REST , API	DPWS,HTTP, CoAP, MQTT

TABLE 4. Types of IoT middlewares.

Middleware	Deployment	Interoperability mechanisms	Scalability	Openness	End-to-end security support
Cloud-based	On cloud	Based on RESTful, API, BLE	Accepts a limited type and number of IoT devices	Supports limited functionalities for users	Weak
Service-based	On the server or cloud	Through device abstraction or based on programming model	Various ranges of IoT devices can be added in the form of services	Computational, not configurable /extendable by users	Weak
Actor-based	All layers of IoT, including sensing devices	Through device abstraction or based on programming model	Devices are exposable and reusable as actors	Users can develop or reuse a pluggable actor	Strong

have been used in IoT architectures to address the high bandwidth issues raised by a large number of terminals, which are accompanied by an increase in the data transmission time [49]. In addition, SOA-IoT can help in the decomposition of complex and monolithic systems into loosely coupled components. A complex system is managed as a set of strongly defined objects or subsystems. When SOA-IoT is applied to IoT, the resulting design can provide extensibility, scalability, modularity, and interoperability among different IoT devices [50].

2) MICROSERVICES

Microservices are a method of splitting large, structured applications into small, extremely decoupled tasks. Furthermore, a separate process is run for each service, rather than full in-memory function calls, and lightweight appliances are used to communicate with each other [51]. The microservice mechanism permits IoT devices to use various messaging protocols inside the service itself. Moreover, separate application features are more capable of independent processing, and allow for resilience of the complete application; in the case of an application crash, only that specific service will be terminated, rather than the entire application. In fact, there is compatibility in several respects between the microservices and the IoT, for instance the use of lightweight communication and software containers to achieve independent software deployment, semi-decentralized management and independent development approaches. Additionally, the conception of choreography in microservices can work as an outline for IoT applications [52].

C. MIDDLEWARE

Middleware is a software layer that serves as a mediator between a different set of applications communicating with a various IoT devices [53]. This subsection briefly overviews the main types of middleware for IoT, namely service-based, cloud-based and actor-based, and presents a comparison in Table 4.

1) SERVICE-BASED

Service-based IoT middleware are deployable in the cloud or on servers. This type of middleware is associated with simple tools such as web applications for viewing the raw data generated by IoT devices. However, the middleware offers limited functionalities to users in the case of composition or integration with other applications or the interoperation of data. Security is ensured by setting up restricted access to protect private and sensitive data. In service-based middleware, the computational units cannot be extended or configured by the end-users. The services are autonomous, dynamically adaptive and can deliver a flexible and simple environment for application development [54]. Service-based middleware requires simultaneous communication between the service consumer and producer [55].

2) CLOUD-BASED

Although cloud-based IoT middleware supports the deployment of only a limited number and type of IoT devices, it easily enables data aggregation and interpretation. In this type of middleware, the functionalities are exposed as a group of APIs with high computation power, massive capacity storage with a service monitoring and analytics tool. A cloud-based IoT service is provided to the device via the cloud, and is controlled by the cloud provider and accessed by the requesting entities. A cloud-based platform is extensively used for several IoT applications, such as smart homes, smart cities and smart energy. It contributes several modules for regular device-to-device activity, for instance device management, data collection, storage management and security.

3) ACTOR-BASED

Actor-based middleware has a relatively large response time and scalability for connecting IoT devices, and hence this type of middleware is deployable in all computation layers, including IoT devices. Furthermore, the middleware allows users to extend the computational units through utilizing or developing pluggable actors. Actor-based IoT middleware does not utilize a particular standard such as RESTful API or BLE to achieve interoperability between IoT devices. However, a specific model of programming or device abstraction is exploited by the middleware to tackle the heterogeneity of IoT devices [55].

D. RUNTIME MANAGEMENT

Runtime management helps to ensure flexible and easy use of the required scripts in the IoT devices, with the aim of providing a service [56]. This section discusses the different types of common runtime management of service management for IoT.

1) COMPOSITION

Service composition enables interaction between IoT entities and consumer requirements [17]. Service composition follows several strategies to select a suitable service and service provider, based on the services that are recognized by the service discovery mechanism. The selection of appropriate services is a crucial task that entails achieving the desired quality and functionality by combining many services to form an integrated composite service. The service composition process involves web suppliers and business processes in IoT.

2) PROVISIONING

Provisioning is the method of preparing and delivering the services of the smart devices to the web. Service provisioning is achieved through collaboration between standard applications, smart and ubiquitous applications. Smart service provisioning offers a new opportunity for conventional internet applications to move toward new ecosystems. A smart object can also be incorporated into the open web standard using web application programming interfaces. The rapid growth of IoT applications has transformed service provisioning from the perspective of always-on services to always-responsive services, i.e. at-runtime responses for any user [8].

3) ORCHESTRATION

Service orchestration supports the integration of multiple services to perform a user task or data synchronization in real time [57]. In the IoT context, orchestration is concerned with the identification of which components or smart devices are needed to form the requested service [11]. An orchestrator can be any IoT device that is used to control the execution transparently to the user. The orchestrator sends a triggering event that checks the condition for carrying out an action using actuators [58]. The development of a service orchestrator requires a deep understanding of service semantics and decomposition of the service request [59].

4) ADAPTION

The IoT model converts objects from conventional to smart objects, to provide the end user with the functions and qualities of the system. IoT systems are rapidly changing, heterogeneous, highly dynamic, and subject to risk and failure. The corresponding system must therefore have the ability to adapt itself at runtime to receive the environment circumstances and transfer the existing business model into a new ecosystem [60].

The adaptation architecture for IoT systems is used for service diagnosis, application diagnosis and service fault recovery [61].

E. SECURITY

Security preservation is a crucial issue for heavily connected devices and spanned services in IoT environments [62]. Service provisioning in IoT orchestrates authentication, authorization, access control and data integrity, to provide secure and satisfactory services to users. This subsection elaborates the various trust and security levels that have been used in service management for IoT.

1) AUTHENTICATION AND AUTHORIZATION

The initial step for a smart device mission is bootstrapping, in which a smart device wakes up and wants to be connected to the ecosystem. The device must first undergo the authentication process before joining the system, to avoid malicious devices joining. If the authentication process is successfully completed, an authorization process, in which the smart device is granted the authorization level necessary to carry out a certain task or service according to predefined policies. Adopting a robust and fine-grained authorization mechanism is crucial to the WoT ecosystem, since smart devices can be discovered easily on the World Wide Web [63]. Most of the existing authentication solutions use a distributed authorization style, in which a back-end server performs complex jobs, requiring rich computing resources. There is typically a server located between the smart device (service provider) and the service requestor, and the service needs the ability to differentiate between the various requests sent by different entities, and to enforce the appropriate authentication decision [64].

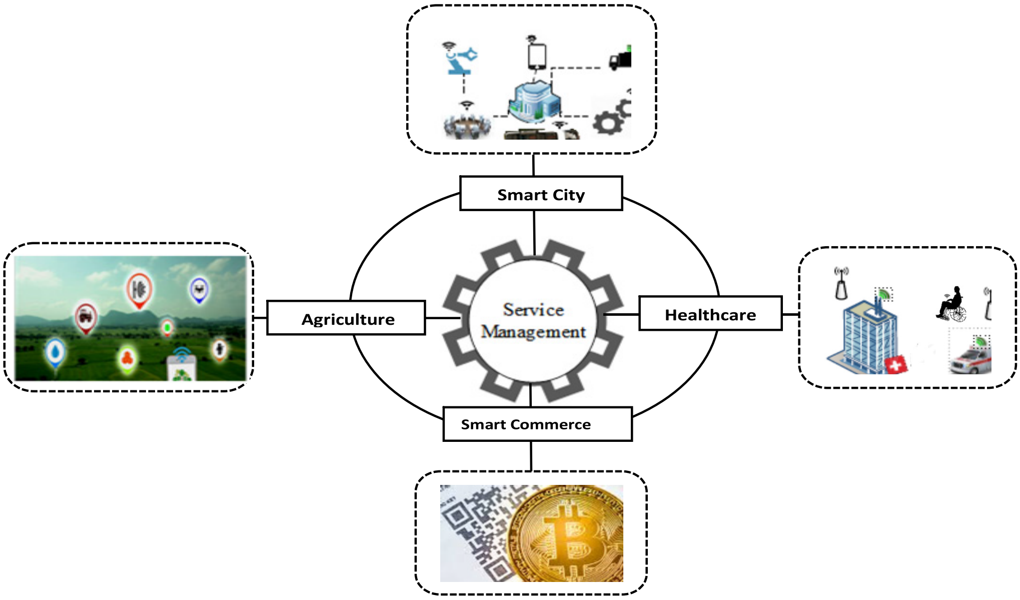

FIGURE 4. Service management domains for IoT.

2) ACCESS CONTROL

The concept of access control is adopted in IoT to protect front-end and back-end data and IoT services and resources by applying data access restrictions. The crucial factor is the method used to allow smart devices to grant access to a service requestor in an IoT ecosystem, where the source of threat can be a malicious device, unauthorized data exposure, or a range of attacks. WoT uses two approaches to enforce access control: the distributed approach and the centralized approach [65]. In the distributed approach, an access control server authenticates an IoT entity and grants it the appropriate access token, which allows access to the IoT resource according to the deployment policy, either permanently or for a specific time interval. In the centralized approach, all the requests pass via an access control server, which issues an authorization status and connects them with the right destination.

3) DATA INTEGRITY AND CONFIDENTIALITY

Secure communication across the WoT is mandatory to preserve data integrity and confidentiality and to thwart attackers [66]. However, encryption solutions require computational capability and memory resources, which cannot be always offered by smart devices. Lightweight end-to-end encryption is established at either the transport layer or the applications layer. Data encryption at the transport layer enables secure communication in the WoT in the form of human-to-Thing and Thing-to-Thing communication. Application-based security concerns with direct interaction and datagram payload data, for instance via application proxies, which are utilized by several firewalls [67].

F. APPLICATIONS

This section discusses examples of IoT applications that rely on service management. Figure 4 shows four of these application domains, namely smart healthcare, smart commerce, smart cities, and smart agriculture.

1) SMART HEALTHCARE

Smart healthcare aims to meet the increasing demand arising from an aging populace with chronic diseases. A smart healthcare system has a sensing layer and service layer. The sensing layer is concerned with acquiring special kinds of health information via sensors and wearable devices [68], while the service layer offers an authentic healthcare service, for instance by processing patient data on health status such as glucose level, heart rate and blood type. Scalability provides the advantages of collecting, processing and analyzing a number of sources of data to obtain feedback, in order to understand the patients' fitness status and the effects of their clinical conditions. Moreover, it leads to the establishing of trust between medical doctors and patients [69].

2) SMART AGRICULTURE

Smart agriculture, also known as precision agriculture, is an IoT application that relies on emerging digital farming technologies such as robots, drones and satellites. Climate changes play a vital role in agriculture sectors, which need to be monitored in terms of weather and the growth of plants and trees. Smart agriculture helps farmers to effectively manage their products and to interact with various stakeholders [70]. Smart agriculture applications enable farmers to monitor irrigation, and to measure the nitrogen, phosphorus,

TABLE 5. Comparison of recent advances in service management for IoT.

Research	Service Management Issues				
	Architectural Organization	Middleware Strategies	Runtime Management	Security	Applications
Min et al. [1]	MS	Actor-based	Composition	N/A	Generic
Alsaryrah et al. [2]	MS	Actor-based		N/A	Generic
Huang [3]	SOA	Cloud-based		N/A	Smart home and smart healthcare
Mohammadi et al. [4]	SOA	Service-based		N/A	Smart city
Sun et al. [6]	MS	Service-based		N/A	Generic
Han and Crespi [8]	MS	Actor-based	Provisioning	Authorization and authentication	Generic
Khan et al. [9]	SOA	Cloud-based		Authorization, Authentication and confidentiality	Smart city
Viejo et al. [12]	SOA	Service-based	Orchestration	Privacy-preserving	Smart city
Wang et al.[13]	SOA	Cloud-based		N/A	Generic
Lee et al. [14]	SOA	Service-based	Adaptation	N/A	Generic

and potassium in liquid manure. Moreover, smart agriculture applications offer opportunities for interaction with other farmers via social networks services. However, security threats to smart agriculture services are increasing due to the impacts of global climate change, and farmers interconnected via social networking need to be trusted in terms of their product suggestions and decision making.

3) SMART COMMERCE

The smart commerce revolution has moved beyond traditional e-commerce models by adding customer-centric, brand-centric, data-centric, and experience-driven models. Among the many IoT services, smart-commerce applications suits the best case for the IoT service. There are two different cases related to smart commerce applications [71]. In the first, when the user notices an advertisement for a product, clicking the ad and enquiring via their website will give information about the product details and provide an option to buy via smart marketing. Furthermore, the selected picture can show the nearest store and offer promotion and discount coupons. In the second case, a consumer's smartphone is used to detect the customer's movements in the supermarket via GPS. This real-time location information helps in analyzing the behavior pattern of the customer in the supermarket. In both cases, smart commerce applications are location-based, personalized and real-time. The main challenges are related to the integration of diverse services and seamless connections between different technologies.

4) SMART CITIES

A smart city is defined as ''a city that engages its citizens and connects its infrastructure electronically'' [72]. Modern and sophisticated sensors provide new opportunities to collect and efficiently use smart city data for urban planning, awareness, policy and decision making. Moreover, managing these data and creating smart services for urban areas requires trust and adoption by various stakeholders, including citizens [9]. Nevertheless, the number of security and privacy challenges from smart city applications are increasing tremendously, such as user privacy related to location, threats to user devices, the hijacking of smart city signals and ransomware attacks on energy management systems.

VI. RECENT ADVANCES IN SERVICE MANAGEMENT FOR IoT

This section critically investigates the existing service management solutions (models, frameworks and protocols). These solutions aim to cope with challenges associated with service provision, orchestration, composition and adaptation, as described in Table 5, and are discussed below.

A. SERVICE COMPOSITION

Min *et al.* [1] implemented the artificial bee colony algorithm for resource and QoS-aware service composition. The authors improved the solution by introducing an operator for resource checking, to ensure that the component services had sufficient resources to execute the task successfully. Resource checking depends on an analysis of the mutual relationships between resources and services. The researcher defined the features of services in different domains, and the implementation showed that the proposed method resulted in considerable improvement. However, the resource checking operator needed to be redesigned for flexible alignment with highly dynamic resources. The actual discovery of domain features and analysis of the impact of each feature on the optimization domain are required.

Alsaryrah *et al.* [2] introduced a pulse algorithm to solve the shortest path optimization problem for a bi-objective optimal balance between energy consumption and QoS in service composition in an IoT environment. The authors deployed several smart objects and ran their algorithm to select the most suitable objects for service composition. The experimental results revealed that the proposed solution required less execution time for various service profiles, achieved considerable performance improvement in terms of energy consumption and network lifetime, and preserved an acceptable QoS.

Huang *et al.* [3] introduced a framework for building smart applications based on intelligent edge computing. This solution pushes the computation process from the cloud server to the edge node, in order to provide reliable and timely data analytics in IoT applications. The proposed framework was implemented with a case study, and a performance comparison was then conducted between running it on an edge node vs. the cloud. The experimental results demonstrated that edge intelligence for IoT services effectively assists in building smart applications and provides situation-awareness and better response time.

Mohammadi *et al.* [4] studied time wastage in the indoor environment of smart city services. The authors proposed a semi-supervised deep reinforcement learning model to satisfy the requirements of indoor localization in smart city applications. The accuracy of the learning agent was improved by setting smart applications to consume both labeled and unlabeled data. The model employed a vibrational auto-encoder as an inference engine to generate optimum policies. The author considered smart buildings as a case study, utilizing Bluetooth technology with low energy signal strength, and applied their model to the problem of indoor localization. The experimental results showed an improvement regarding the distance to the target, and better performance than deep reinforcement learning model.

Sun *et al.* [6] introduced a two-tier framework to represent the functionalities of smart Things in an IoT service. The authors utilized heuristic algorithms to facilitate the coordination of Smart Things for service composition in scenarios where the task requirements span across multiple Things. The ant colony, genetic, and swarm algorithms were adopted as heuristic methods of finding the optimal service compositions. The experimental results showed that the adopted algorithms find the approximately optimal service composition while reducing the energy consumption and prolonging the network lifetime.

B. SERVICE PROVISIONING

Han and Crespi [8] proposed an architecture to support service provisioning for smart objects, the architecture associated with the semantic annotation. The authors aim to enable the seamless integration of IoT applications with the web through setting smart objects as IP-based entities that can ensure low energy consumption and serve as an integral part of the online service. The researchers assessed their solution empirically using several prototypes and applications. The results demonstrated that the proposed architecture supports the integration of IoT applications on the Web. However, further testing on the different scenario of WoT is required to verify the efficiency of the proposed architecture.

Khan *et al.* [9] proposed a service provisioning framework to ensure security and privacy during service provisioning in a smart city. This solution achieved its goal by overcoming the problems of service compromise, malicious citizens, and malicious service providers. The system meets several requirements of end-to-end service provisioning, such as trust-based data acquisition, secure processing, transmission, and preservation of service integrity with legitimate provisioning. The authors tested their framework in different scenarios of service provisioning for smart cities, and developed a lightweight communication protocol for verification purposes. The results proved the usefulness of the framework based on the individual components, but the robustness of this protocol not verified in a real smart city environment.

C. SERVICE ORCHESTRATION

Viejo and Sánchez [11] proposed privacy-by-design protocols for service orchestration and delivery in fog-enabled IoT systems. Their solution secures data exchange in the network by employing attribute-based encryption in fog-based IoT, and ensures that the data necessary to satisfy the service are released only to the only entities involved, thus satisfying the data minimization principle. The proposed protocols run in any IoT architecture consist of fog nodes, sensing devices and the cloud. The experimental results showed that the proposed protocol was secure and feasible, although the protocol was not evaluated in a realistic, operational IoT environment.

Wang *et al.* [12] introduced a linear programming model together with an optimization algorithm for a performance and resource-aware orchestration system. The proposed model was aimed at ensuring performance maximization while optimizing resource utilization in the IoT environment in the presence of large volumes of traffic. The authors built a prototype for implementing their solution on OpenStack, and the experimental results revealed that the proposed model obtained better performance than the existing solutions.

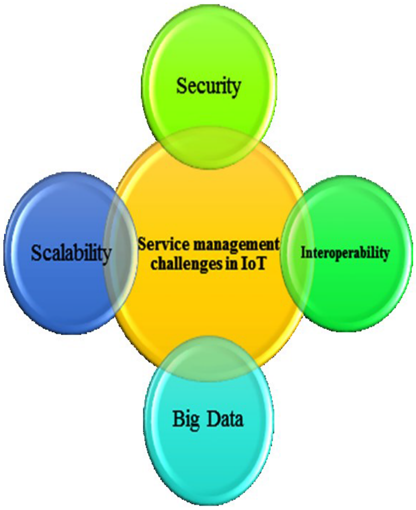

FIGURE 5. Service management challenges in IoT.

D. SERVICE ADAPTION

Lee *et al.* [13], proposed a model for tackling the problem of unsatisfactory provisioning of smart IoT services. The solution introduced the service composition of two layers, namely the user control layer and the cloud control layer; the former is concerned with the management of service context awareness and end-to-end service connection, and the latter with the management of service profiling, resource allocation, service scheduling, and adaptation policies. The authors conducted experiments on a lightweight prototype. In the experimental results, the proposed model showed performance improvement in terms of throughput in comparison with the legacy binding approach. However, investigation and consideration of advanced issues such as mobility management are needed for the effective utilization of the proposed model.

VII. OPEN RESEARCH CHALLENGES

Current service management approaches for IoT have the potential to provide numerous solutions, but many challenges have not yet been fully addressed and require collaboration from standardization committees, hardware manufacturers, software developers and IoT stakeholders. This section discusses several challenges related to service management in the context of IoT, as shown in Figure 5.

A. INTEROPERABILITY-RELATED CHALLENGES

Service management in IoT encounters three types of interoperability challenges: connectivity, semantic and syntactic.

1) CONNECTIVITY CHALLENGES

These are concerned with enabling seamless integration and information exchange between IoT systems with diverse device capabilities, via different networking technologies, standards, communication protocols and platforms [73]. A lack of connectivity and interoperability leads to a limited ability to integrate diverse devices into the different service management platforms of IoT systems.

2) SEMANTIC CHALLENGES

These are concerned with the ability of various IoT applications and services to interpret the exchanged data in a meaningful way [74]. A high level of semantic incompatibility between data and information models in IoT systems leads to different descriptions of resource and operational procedures, which results in failure of the system [75].

3) SYNTACTIC CHALLENGES

These are concerned with the data format and structure that is used in a service or information exchange between heterogeneous IoT entities and systems. Syntactic interoperability can be achieved by defining an interface for each IoT resource and exposing its metadata to the relevant entities. The challenge arises when the encoding rule used by the information sender is different from the decoding rule used by the receiver, which results in message mismatching.

Tackling interoperability challenges in IoT requires cross-domain interoperability solutions.

B. SCALABILITY-RELATED CHALLENGES

Service management in IoT is expected to encounter several challenges related to vertical scalability and horizontal scalability.

1) VERTICAL SCALABILITY CHALLENGES

These are related to the ability to support additions to enhance the capability of an IoT device and to updating of firmware and software applications running on sensing nodes and IoT gateways. The challenges are associated with the difficulty of keeping track of which updates are available, and consistently applying updates across a network containing heterogeneous components that communicate via a set of various protocols.

2) HORIZONTAL SCALABILITY CHALLENGES

These are concerned with the addition of new devices, software and services, which need scalable service registry and harmonic interaction. The horizontal scalability of service management systems in IoT is associated with many challenges related to networking protocols, security, privacy, fault tolerance, access control, trust and governance. The problem arises when the service management system fails to integrate a wide range of new IoT devices [76].

To accommodate rapid scalability in an IoT system, the service management system needs to achieve the performance improvement of IoT applications, in order to provide high QoS for a scaled-up version of the IoT system.

C. SECURITY-RELATED CHALLENGES

Conventional security solutions and practices cannot handle the expansion, mobility, resource constraints and new security requirements of IoT. The main type of trust and security challenges in IoT are identity management, authentication and authorization, and vulnerability detection.

1) IDENTITY MANAGEMENT CHALLENGES

These involve a lack of ability to discover and manage the identities of IoT entities across the different integrated IoT-based applications to support authentication techniques [77].

2) AUTHENTICATION AND AUTHORIZATION

These are crucial issues for ensuring the security of IoT systems in terms of supporting access control and preserving privacy and integrity. Authorization aims to determine which services, resources and apps each device can access in the system. The challenge arises when IoT devices fail to perform authentication due to weak password authentication or poor password management policies [78].

3) VULNERABILITY DETECTION

This is concerned with monitoring IoT activity logs for anomalies, engaging in penetration testing to expose vulnerabilities, and the identification and notification of security threats. Challenges arise that are related to high scalability in terms of the number and diversity of devices, apps, communication protocols and services, which makes it difficult to identify vulnerabilities.

4) AVAILABILITY

These are associated with maintaining the durability and accessibility of IoT services. Several inherited and emerging availability problems arise in IoT, such as device failure and dis-connectivity. The challenges are related to the protection of IoT entities against physical tampering and denial of service attacks.

In the future, reliable and scalable security mechanisms, protocols, polices, practices are needed for IoT. Moreover, technologies are needed to provide advanced control of data exchanges among IoT entities.

D. BIG DATA-RELATED CHALLENGES

Some IoT systems handle massive data at the cloud, device or IoT gateway level. In IoT, big data processing encounters several challenges related to accuracy, real-time analytics, and visualization [79].

1) ACCURACY CHALLENGES

A service management system is expected to provide an effective data analytics technique for extracting knowledge from the massive data generated by IoT entities. The challenge is to extract information from the heterogeneous and complex data generated by different IoT entities [80].

2) REAL-TIME ANALYTICS CHALLENGES

Many IoT applications require data extraction and processing in real time. Analyzing data in volumes measured in terabytes or petabytes in real time is associated with several challenges such as data integration and visualization.

3) VISUALIZATION CHALLENGES

Visualization is an auxiliary for big data analytics that involves dashboard and mobile apps. Seamless synchronization between visualization and the data analytics process allows the results of the analysis of IoT applications to be meaningful and understandable. The challenge is to generate a visual representation of highly heterogeneous big data.

VIII. CONCLUSION

IoT is the largest community in cyberspace, comprising billions of heterogeneous computation and communication devices, that is intended to provide a wide range of services. This environment needs careful device synchronization, proper scalability and interoperability management, service monitoring and QoS measurement. These tasks cannot be performed without an effective service management system, and this article therefore focuses on service management for IoT. First, we investigated the recent advances in the literature related to service management issues for IoT. We then determined the requirements of service management for IoT, and provided a thematic taxonomy. Next, we presented several opportunities related to service management in IoT, and identified open research challenges that can act as a guide for future research. Finally, we conclude that current service management solutions for IoT face challenging issues that must be addressed in the future.

REFERENCES

[1] X. Min, X. Xu, Z. Liu, D. Chu, and Z. Wang, "An approach to resource and QoS-aware services optimal composition in the big service and Internet of Things," *IEEE Access*, vol. 6, pp. 39895–39906, 2018.

[2] O. Alsaryrah, I. Mashal, and T.-Y. Chung, "Bi-objective optimization for energy aware Internet of Things service composition," *IEEE Access*, vol. 6, pp. 26809–26819, 2018.

[3] Z. Huang, K.-J. Lin, B.-L. Tsai, S. Yan, and C.-S. Shih, "Building edge intelligence for online activity recognition in service-oriented IoT systems," *Future Gener. Comput. Syst.*, vol. 87, pp. 557–567, Oct. 2018.

[4] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J. Oh, "Semisupervised deep reinforcement learning in support of IoT and smart city services," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 624–635, Apr. 2018.

[5] Y. Qin, Q. Z. Sheng, N. J. G. Falkner, S. Dustdar, H. Wang, and A. V. Vasilakos, "When things matter: A survey on data-centric Internet of Things," *J. Netw. Comput. Appl.*, vol. 64, pp. 137–153, Apr. 2016.

[6] M. Sun, Z. Shi, S. Chen, Z. Zhou, and Y. Duan, "Energy-efficient composition of configurable Internet of Things services," *IEEE Access*, vol. 5, pp. 25609–25622, 2017.

[7] P. Spiess, S. Karnouskos, D. Guinard, D. Savio, O. Baecker, L. M. S. de Souza, and V. Trifa, "SOA-based integration of the Internet of Things in enterprise services," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jul. 2009, pp. 968–975.

[8] S. N. Han and N. Crespi, "Semantic service provisioning for smart objects: Integrating IoT applications into the Web," *Future Gener. Comput. Syst.*, vol. 76, pp. 180–197, Nov. 2017.

[9] Z. Khan, Z. Pervez, and A. G. Abbasi, "Towards a secure service provisioning framework in a smart city environment," *Future Generat. Comput. Syst.*, vol. 77, pp. 112–135, Dec. 2017.

[10] D. Salikhov, K. Khanda, K. Gusmanov, M. Mazzara, and N. Mavridis, "Microservice-based IoT for smart buildings," 2016, *arXiv:1610.09480*. [Online]. Available: https://arxiv.org/abs/1610.09480

[11] A. Viejo and D. Sánchez, "Secure and privacy-preserving orchestration and delivery of fog-enabled IoT services," *Ad Hoc Netw.*, vol. 82, pp. 113–125, Jan. 2019.

[12] J. Wang, H. Qi, K. Li, and X. Zhou, "PRSFC-IoT: A performance and resource aware orchestration system of service function chaining for Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1400–1410, Jun. 2018.

[13] T.-D. Lee, B. M. Lee, and W. Noh, "Hierarchical cloud computing architecture for context-aware IoT services," *IEEE Trans. Consum. Electron.*, vol. 64, no. 2, pp. 222–230, May 2018.

[14] T. Riedel, N. Fantana, A. Genaid, D. Yordanov, H. R. Schmidtke, and M. Beigl, "Using Web service gateways and code generation for sustainable IoT system development," in *Proc. IEEE Internet Things (IOT)*, Nov./Dec. 2010, pp. 1–8.

[15] R. K. Kodali, V. Jain, and S. Karagwal, "IoT based smart greenhouse," in *Proc. IEEE Region Humanitarian Technol. Conf. (R10-HTC)*, vol. 10, Dec. 2016, pp. 1–6.

[16] H. Cervantes and R. S. Hall, "Automating service dependency management in a service-oriented component model," in *Proc. ICSE CBSE Workshop*, 2003, pp. 1–6.

[17] P. Asghari, A. M. Rahmani, and H. H. S. Javadi, "Service composition approaches in IoT: A systematic review," *J. Netw. Comput. Appl.*, vol. 120, pp. 61–77, Oct. 2018.

[18] M. Rosen, B. Lublinsky, K. T. Smith, and M. J. Balcer, *Applied SOA: Service-Oriented Architecture and Design Strategies*. Hoboken, NJ, USA: Wiley, 2012.

[19] D. Georgakopoulos and M. Papazoglou, *Service-Oriented Computing*. Cambridge, MA, USA: MIT Press, 2009.

[20] M. Thiyagarajan and C. Raveendra, "Role of Web service in Internet of Things," in *Proc. IEEE 3rd Int. Conf. Appl. Theor. Comput. Commun. Technol. (iCATccT)*, Dec. 2017, pp. 268–270.

[21] D. Austin, A. Barbir, C. Ferris, and S. Garg. (Oct. 25, 2004). *Web Services Architecture Requirements*. [Online]. Available: https://www.w3.org/TR/2004/NOTE-wsa-reqs-20040211/

[22] M. Kovatsch, K. Kajimoto, and U. Davuluru. (Oct. 25, 2017). *Web of Things (WoT) Architecture (W3C First Public Working*. Accessed: Sep. 14, 2017. [Online]. Available: https://www.w3.org/TR/wot-architecture/

[23] D. Hibbert, *Thunder Cloud: Managing Reward in a Digital Age*. Bloomington, IN, USA: AuthorHouse, 2016.

[24] Salesforce. (Sep. 18, 2018). *Salesforce IoT*. [Online]. Available: https://www.salesforce.com/uk/products/iot-cloud/overview/.

[25] AW Services. (Sep. 17, 2018). *AWS IoT*. [Online]. Available: https://aws.amazon.com/iot/

[26] S. Bhatt, F. Patwa, and R. Sandhu, "Access control model for AWS Internet of Things," in *Proc. Int. Conf. Netw. Syst. Secur.* Cham, Switzerland: Springer, 2017, pp. 721–736.

[27] G. Cloud. (Sep. 20, 2018). *Google Cloud IoT*. [Online]. Available: https://cloud.google.com/solutions/iot/

[28] Microsoft. (Sep. 30, 2018). *Azure IoT Hub*. [Online]. Available: https://azure.microsoft.com/en-us/services/iot-hub/

[29] Cisco. (Oct. 25, 2018). *Cisco IoT*. [Online]. Available: https://www.cisco.com/c/en/us/solutions/internet-of-things/overview.html

[30] AT&T. (Sep. 17, 2018). *AT&T IoT Platform Services*. [Online]. Available: https://iotplatform.att.com/

[31] IBM. (Nov. 6, 2018). *IBM Watson IoT Platform*. [Online]. Available: https://www.ibm.com/my-en/marketplace/internet-of-things-cloud

[32] Oracle. (Sep. 30, 2018). *Oracle Internet of Things Cloud Service*. [Online]. Available: https://docs.oracle.com/en/cloud/paas/iot-cloud/index.html

[33] UBIDOTS. (Sep. 30, 2018). *IoT and Cloud Tools to Build Your Business*. [Online]. Available: https://ubidots.com/platform/

[34] A. Taherkordi and F. Eliassen, "Scalable modeling of cloud-based IoT services for smart cities," in *Proc. PerCom Workshops*, 2016, pp. 1–6.

[35] A. Geraci, F. Katki, L. McMonegal, B. Meyer, and H. Porteous, *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*, IEEE Standard 610-1990, 1991, pp. 9–13.

[36] J. Delgado, "Service interoperability in the Internet of Things," in *Internet of Things and Inter-Cooperative Computational Technologies for Collective Intelligence*. Heidelberg, Germany: Springer, 2013, pp. 51–87.

[37] R. Kyusakov, "Efficient WEB services for end-to-end interoperability of embedded systems," Ph.D. dissertation, Dept. Comput. Sci., Embedded Internet Syst. Lab., Luleå Tekniska Univ., Luleå, Sweden, 2014.

[38] F. Leymann, "Web services: Distributed applications without limits," in *Proc. Database Syst. Bus., Technol. Web BTW*, vol. 26, 2003, pp. 2–23.

[39] S. Mubeen, S. A. Asadollah, A. V. Papadopoulos, M. Ashjaei, H. Pei-Breivold, and M. Behnam, "Management of service level agreements for cloud services in IoT: A systematic mapping study," *IEEE Access*, vol. 6, pp. 30184–30207, 2018.

[40] M. I. Ladan, "Web services metrics: A survey and a classification," *J. Commun. Comput.*, vol. 9, no. 7, pp. 824–829, 2012.

[41] D. Kyriazis, "Cloud computing service level agreements-exploitation of research results," Eur. Commission Directorate Gen. Commun. Netw. Content Technol. Unit, Brussels, Belgium, Tech. Rep. 5, 2013, p. 29.

[42] D. Guinard, "Towards opportunistic applications in a Web of Things," in *Proc. 8th IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PERCOM Workshops)*, Mar./Apr. 2010, pp. 863–864.

[43] E. Ahmed, I. Yaqoob, I. A. T. Hashem, I. Khan, A. I. A. Ahmed, M. Imran, and A. V. Vasilakos, "The role of big data analytics in Internet of Things," *Comput. Netw.*, vol. 129, pp. 459–471, Dec. 2017.

[44] X. Liu, M. Zhao, S. Li, F. Zhang, and W. Trappe, "A security framework for the Internet of Things in the future Internet architecture," *Future Internet*, vol. 9, no. 3, p. 27, 2017.

[45] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef, "Internet of Things security: A top-down survey," *Comput. Netw.*, vol. 141, pp. 199–221, Aug. 2018.

[46] J. Gao, F. Liu, H. Ning, and B. Wang, "RFID coding, name and information service for Internet of Things," in *Proc. IET Conf. Wireless, Mobile Sensor Netw. (CCWMSN)*, 2007, pp. 36–39.

[47] M. Gigli and S. Koo, "Internet of Things: Services and applications categorization," *Adv. Internet Things*, vol. 1, no. 2, pp. 27–31, 2011.

[48] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the SOA-based Internet of Things: Discovery, query, selection, and on-demand provisioning of Web services," *IEEE Trans. Services Comput.*, vol. 3, no. 3, pp. 223–235, Jul./Sep. 2010.

[49] W.-C. Chien, C.-F. Lai, H.-H. Cho, and H.-C. Chao, "A SDN-SFC-based service-oriented load balancing for the IoT applications," *J. Netw. Comput. Appl.*, vol. 114, pp. 88–97, Jul. 2018.

[50] C. M. Sosa-Reyna, E. Tello-Leal, and D. Lara-Alabazares, "Methodology for the model-driven development of service oriented IoT applications," *J. Syst. Archit.*, vol. 90, pp. 15–22, Oct. 2018.

[51] M. Joneja, "Microservices as a design choice for IoT," in *Proc. Conf.*, 2016, pp. 1–3.

[52] B. Butzin, F. Golatowski, and D. Timmermann, "Microservices approach for the Internet of Things," in *Proc. IEEE 21st Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2016, pp. 1–6.

[53] S. M. Balakrishnan and A. K. Sangaiah, "MIFIM—Middleware solution for service centric anomaly in future Internet models," *Future Gener. Comput. Syst.*, vol. 74, pp. 349–365, Sep. 2017.

[54] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 3, no. 1, pp. 70–95, Feb. 2016.

[55] A. Farahzadi, P. Shams, J. Rezazadeh, and R. Farahbakhsh, "Middleware technologies for cloud of things: A survey," *Digit. Commun. Netw.*, vol. 4, no. 3, pp. 176–188, Aug. 2018.

[56] M. Kovatsch, M. Lanter, and S. Duquennoy, "Actinium: A RESTful runtime container for scriptable Internet of Things applications," in *Proc. IEEE 3rd Int. Conf. Internet Things (IOT)*, Oct. 2012, pp. 135–142.

[57] G. Fortino and P. Trunfio, Eds., *Internet of Things Based on Smart Objects: Technology, Middleware and Applications*. Cham, Switzerland: Springer, 2014, pp. 85–105.

[58] L. Bergesio, A. M. Bernardos, and J. R. Casar, "An object-oriented model for object orchestration in smart environments," *Procedia Comput. Sci.*, vol. 109, no. 109C, pp. 440–447, 2017.

[59] E. Chindenga, M. S. Scott, and C. Gurajena, "Semantics based service orchestration in IoT," in *Proc. ACM South Afr. Inst. Comput. Sci. Inf. Technol.*, 2017, Art. no. 7.

[60] M. Hussein, S. Li, and A. Radermacher, "Model-driven development of adaptive IoT systems," in *Proc. 4th Int. Workshop Interplay Model-Driven Component-Based Softw. Eng. (ModComp)*, 2017, p. 20.

[61] X. Peng and B. Pernici, "A service diagnosis and adaptation scheme in service-oriented IoT," in *Proc. Res. IEEE 2nd Int. Forum Technol. Soc. Ind. Leveraging Better Tomorrow (RTSI)*, Sep. 2016, pp. 1–6.

[62] Z. A. Khan, "Using energy-efficient trust management to protect IoT networks for smart cities," *Sustain. Cities Soc.*, vol. 40, pp. 1–15, Jul. 2018.

[63] S. Gerdes, O. Bergmann, and C. Bormann, *Delegated CoAP Authentication and Authorization Framework (DCAF)*, document IETF draftgerdes-core-dcaf-authorize-02, 2014.

[64] S. Gerdes, O. Bergmann, and C. Bormann, "Delegated authenticated authorization for constrained environments," in *Proc. IEEE 22nd Int. Conf. Netw. Protocols (ICNP)*, Oct. 2014, pp. 654–659.

[65] S. El Jaouhari, A. Bouabdallah, and J.-M. Bonnin, "Security issues of the Web of Things," in *Managing the Web of Things*. Amsterdam, The Netherlands: Elsevier, 2017, pp. 389–424.

[66] J. Granjal, E. Monteiro, and J. S. Silva, "On the feasibility of secure application-layer communications on the Web of Things," in *Proc. 37th Annu. IEEE Conf. Local Comput. Netw.*, Oct. 2012, pp. 228–231.

[67] J. Granjal, E. Monteiro, and J. S. Silva, "Application-layer security for the WoT: Extending CoAP to support end-to-end message security for Internet-integrated sensing applications," in *Proc. Int. Conf. Wired/Wireless Internet Commun.*, Berlin, Germany: Springer, 2013, pp. 140–153.

[68] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare," *Future Generat. Comput. Syst.*, vol. 78, pp. 659–676, Jan. 2018.

[69] E. Ahmed, I. Yaqoob, A. Gani, M. Imran, and M. Guizani, "Internet-of-Things-based smart environments: State of the art, taxonomy, and open research challenges," *IEEE Wireless Commun.*, vol. 23, no. 5, pp. 10–16, Oct. 2016.

[70] O. Westermann, W. Förch, P. Thornton, J. Körner, L. Cramer, and B. Campbell, "Scaling up agricultural interventions: Case studies of climate-smart agriculture," *Agricult. Syst.*, vol. 165, pp. 283–293, Sep. 2018.

[71] X. Shang, R. Zhang, and Y. Chen, "Internet of Things IoT service architecture and its application in e-commerce," *J. Electron. Commerce Org.*, vol. 10, no. 3, pp. 44–55, 2012.

[72] S. Musa, "Smart cities—A road map for development," *J. Telecommun. Syst. Manage.*, vol. 5, no. 3, pp. 19–23, 2016.

[73] O. Hahm, E. Baccelli, H. Petersen, and N. Tsiftes, "Operating systems for low-end devices in the Internet of Things: A survey," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 720–734, Oct. 2016.

[74] M. Ushold, C. Menzel, and N. Noy, "Semantic integration & interoperability using RDF and OWF," W3C Editor's Draft, Cambridge, MA, USA, Tech. Rep. 3, 2005.

[75] P. Murdock *et al.*, "Semantic interoperability for the Web of Things," Telecom Sudparis Paris Saclay, Paris Saclay Univ., Paris, France, Tech. Rep. 18, 2016.

[76] I. Yaqoob, E. Ahmed, I. A. T. Hashem, A. I. A. Ahmed, A. Gani, M. Imran, and M. Guizani, "Internet of Things architecture: Recent advances, taxonomy, requirements, and open challenges," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 10–16, Jun. 2017.

[77] C. Chibelushi, A. Eardley, and A. Arabo, "Identity management in the Internet of Things: The role of MANETs for healthcare applications," *Comput. Sci. Inf. Technol.*, vol. 1, no. 2, pp. 73–81, 2013.

[78] M. Ammar, G. Russello, and B. Crispo, "Internet of Things: A survey on the security of IoT frameworks," *J. Inf. Secur. Appl.*, vol. 38, pp. 8–27, Feb. 2018.

[79] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa, and I. Yaqoob, "Big IoT data analytics: Architecture, opportunities, and open research challenges," *IEEE Access*, vol. 5, pp. 5247–5261, 2017.

[80] R. A. A. Habeeb, F. Nasaruddin, A. Gani, I. A. T. Hashem, E. Ahmed, and M. Imran, "Real-time big data processing for anomaly detection: A survey," *Int. J. Inf. Manage.*, vol. 45, pp. 289–307, Apr. 2019.

ABDELMUTTLIB IBRAHIM ABDALLA AHMED received the B.Sc. degree in computer science from OIU, Sudan, and the M.S. degree in computer science from IIUI, Pakistan. He is currently pursuing a Ph.D. degree with the University of Malaya, Malaysia. He has published several high-impact research articles in reputed international journals and conferences. His research interest areas include trust and reputation systems, the Internet of Things, cloud computing, vehicular communication, and software-defined networks.

ABDULLAH GANI received the B.Phil. and M.Sc. degrees in information management from the University of Hull, U.K., and the Ph.D. degree in computer science from the University of Sheffield, U.K. Prior to his degree studies, he acquired the Teaching Certificate from the Kinta Teaching College, Ipoh, and the Diploma degree in computer science from ITM. He has vast teaching experience due to having worked in a number of educational institutions locally and abroad—schools, the Malay Women Teaching College, Melaka, Ministry of Education; the Rotterham College of Technology and Art, Rotterham, U.K.; and the University of Sheffield, U.K. He is currently a Professor with the Dean Faculty of Computing and Informatics, Univesiti Malaysia Sabah, and also an Honory Professor with the Department of Computer System and Technology, Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. He is also working on mobile cloud computing, big data, and the IoT. He is also the Dean of the Faculty of Computing and Informatics, Universiti of Malaysia Sabah. His interest in research kicked off in 1983 when he was chosen to attend the 3-Month Scientific Research Course in RECSAM, Ministry of Education, Malaysia. His current research interests include self-organized systems, machine learning, reinforcement learning, and wireless related networks. He was elected as a Fellow of the Academy of Sciences Malaysia (FASc) for Engineering and Computer Science Discipline.

SITI HAFIZAH AB HAMID received the B.S. degree (Hons.) in computer science from the University of Technology, Malaysia, the M.S. degree in computer system design from The University of Manchester, U.K., and the Ph.D. degree in computer science from the University of Malaya, Malaysia. She is currently an Associate Professor with the Department of Software Engineering, Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. She has authored more than 75 research articles in different fields, including mobile cloud computing, big data, software engineering, machine learning, and the IoT.

ABDELZAHIR ABDELMABOUD received the M.Sc. degree in computer science and information from Gezira University, Sudan, and the Ph.D. degree in software engineering from Universiti Teknologi Malaysia (UTM), Malaysia. He has worked experience for more than 20 years in both industry and academia. He is currently an Assistant Professor with the Department Information Systems, College of Science and Arts, King Khalid University, Muhayil Asir, Saudi Arabia. He is also a member of the Software Engineering Research Group (SERG), UTM. His research interests include cloud computing, the Internet of Things, computer visioning, and information security.

HASSAN JAMIL SYED received the B.E. degree in electrical engineering from the Quaid-e-Awam University of Engineering, Science, and Technology, Nawabshah, Pakistan, in 2003, the master's degree in personal mobile and satellite communication from the University of Bradford, England, U.K in 2008, and the Ph.D. degree in computer science from the University of Malaya, Malaysia in 2019. He was with WorldCall Telecom Ltd., Karachi, Pakistan, as a Network Engineer for two years, after completing the master's degree. He has four years of teaching experience. He was a full-time Assistant Professor with the Faculty of Engineering Science and Technology, Iqra University, Karachi, Pakistan. He is currently serving as an Assistant Professor with the Department of Computer Sciences, National University of Computer and Emerging Sciences, Karachi, Pakistan. During his career, he supervised several final-year projects and taught courses in the Electronics, Telecommunication, and Computer Science Departments. His research interests include performance analysis, stability, reliability, scalability of the cloud, and the Internet of Things.

RIYAZ AHAMED ARIYALURAN HABEEB MOHAMED received the B.E. degree in computer science from Sathyabama University, Chennai, in 2008, the master's degree in software engineering from the University of Malaya, in 2013, and the Ph.D. degree from the Faculty of Computer Science and Information Technology (FSKTM), University of Malaya. He was a Software Engineer, working for clients such as Airbus, Liebherr, Pratt & Whitney prior to his journey into the education industry as an academician. He is currently a fulltime Senior Lecturer with the Faculty of Science Technology Engineering and Mathematics (FSTEM), International University of Malaya-Wales. He has published several research articles in international journals and conferences. His current research interests include real-time, deep learning, big data, machine learning, and cloud computing.

IHSAN ALI received the M.S. degree in computer system engineering from the GIK Institute, in 2008. He is currently pursuing the Ph.D. degree with the Faculty of Computer Science and Information Technology, University of Malaya. He is currently an active Research Associate with the Center for Mobile Cloud Computing Research (C4MCCR), Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia. He has published more than 40 high impact research journal articles including a highly reputable the *IEEE Communication Magazine*. He has been actively involved in research and teaching activities for the last ten years in different country including Saudi Arabia, USA, Pakistan, and Malaysia. His research interests include wireless sensor networks, robotics in WSNS, sensor cloud, fog computing, the IoT, ML/DL in wireless sensor networks. He has served as a Technical Program Committee Member for several well-known conferences including IWCMC 2017-2018, AINIS 2017, Future 5V 2017, ICACCI-2018, INAIT 2019, DiCES-N19, CCNC2020, ICCAIS2020, and CSNT2020 and also an Organizer of the Special Session on fog computing in Future 5V 2017. He is also an Active Reviewer of *Computers and Electrical Engineering*, *KSII Transactions on Internet and Information Systems*, *Mobile Networks and Applications*, the *International Journal of Distributed Sensor Networks*, *Journal of Advanced Transportation*, the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, *Computer Networks*, IEEE ACCESS, *Wireless Communications and Mobile Computing*, and the *IEEE Communication Magazine*.