Mobile Networks and Applications
https://doi.org/10.1007/s11036-018-1089-9

Interoperability in Internet of Things: Taxonomies and Open Challenges

Mahda Noura[1] · Mohammed Atiquzzaman[2] · Martin Gaedke[1]

© The Author(s) 2018

Abstract

In the last few years, many smart objects found in the physical world are interconnected and communicate through the existing internet infrastructure which creates a global network infrastructure called the Internet of Things (IoT). Research has shown a substantial development of solutions for a wide range of devices and IoT platforms over the past 6-7 years. However, each solution provides its own IoT infrastructure, devices, APIs, and data formats leading to interoperability issues. Such interoperability issues are the consequence of many critical issues such as vendor lock-in, impossibility to develop IoT application exposing cross-platform, and/or cross-domain, difficulty in plugging non-interoperable IoT devices into different IoT platforms, and ultimately prevents the emergence of IoT technology at a large-scale. To enable seamless resource sharing between different IoT vendors, efforts by several academia, industry, and standardization bodies have emerged to help IoT interoperability, i.e., the ability for multiple IoT platforms from different vendors to work together. This paper performs a comprehensive survey on the state-of-the-art solutions for facilitating interoperability between different IoT platforms. Also, the key challenges in this topic is presented.

Keywords Internet of Things · Interoperability · IoT platforms · Survey

1 Introduction

The term Internet of Things (IoT), first coined by Kevin Ashton around 1999 [1], has recently been an emerging technology in a broad range of domains. IoT is defined as the connection of physical things ("objects") and places via the Internet [2, 3]. This vision defines a technological revolution where physical and virtual things would be connected to other things and to the current Internet infrastructure. According to the European Research Cluster on the Internet of Things (IERC) [4], IoT is defined as: "a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual things have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network". An abundance of smart connected devices and platforms have been integrated in a wide range of applications like commerce, healthcare, agriculture, utilities, energy, transportation, industrial control and buildings, etc. [5].

Not surprisingly, big vendors like Amazon[1](AWS IoT), Cisco[2] (Jasper), IBM[3] (Watson), Apple[4] (HomeKit), Google[5] (Brillo), Microsoft[6] (Azure IoT), and Qualcomm[7] (AllJoyn) have rapidly proliferated in the IoT market in the last few years. Besides, the European project Unify-IoT [6], lately identified that there are more than 300 IoT platforms in the current market, and more to come. Each of these platforms promotes its own IoT infrastructure, proprietary protocols and interfaces, incompatible standards, formats, and semantics which creates closed ecosystems (sometimes called *stove pipes* or *silos*). Nevertheless, the necessity for these different solutions to seamlessly work together, i.e. *IoT interoperability*, is growing. A new McKinsey analysis [7] points out a

✉ Mahda Noura
mahda.noura@informatik.tu-chemnitz.de

Mohammed Atiquzzaman
atiq@ou.edu

Martin Gaedke
martin.gaedke@informatik.tu-chemnitz.de

[1] Technische Universität Chemnitz, Chemnitz, Germany

[2] University of Oklahoma, Norman, OK 73109, USA

[1] www.amazon.com/iot
[2] www.jasper.com
[3] www.ibm.com/watson
[4] www.apple.com/lae/ios
[5] https://developers.google.com/weave
[6] https://azure.microsoft.com
[7] https://developer.qualcomm.com/software/alljoyn

Published online: 21 July 2018

substantial threat to the predicted economic value: *missing interoperability*. Particularly, the authors state that 40% of the potential benefits of IoT can be obtained with the interoperability between IoT systems.

From the point view of the IoT providers', lack of interoperability means that service providers are bound to the IoT device or software offered by a single provider and must stick with it, which may bring the potential risk of higher operation cost later on, as well as product functionality and stability issues [8]. The incompatibility between different IoT platforms helps to protect the environment of IoT platform providers temporarily until the IoT market develops more mature. In particular, it is very costly for small companies to support heterogenous interfaces of all diverse platforms.

From the perspective of application developers, incompatibility between IoT platforms results in adapting their application to the platform specific API and information models of each different platform, which prevents cross-platform, *i.e. applications which operate on multiple platforms* and cross-domain application development, *i.e. applications which combine different domains*.

The importance of the interoperability challenge in IoT has been emphasized by both academia and industry. The industry attempts to address IoT interoperability challenges through *standardization*. Several efforts have emerged to establish standards for providing interoperability between IoT devices, networks, services, data formats owned by different providers. The European Union has also recently funded several research projects under the H2020 program focusing on the federation of IoT platforms. However, it may take a long time before the related standards are fully agreed upon and accepted, if ever. To resolve this issue, researchers in both academia and industry have been developing a list of innovative solutions for interoperability and heterogeneity in different IoT systems.

To help readers understand the status and future trends of IoT interoperability, we reviewed the past, present and future developments related to enabling technologies and solutions for addressing interoperability. This paper can make IoT experts become more aware of the challenges and opportunities that are in this increasingly crucial topic and bring their proficiency to aid solving research challenges for providing interoperability between services, application, and platforms in IoT. It is important to note that this article is an extended version of the conference paper published in the "Internet of Things as a service" [9], which includes the following contributions extended:

1) A more detailed taxonomy for IoT interoperability.
2) A deep insight into the state-of-the-art, including ongoing projects and research dealing with IoT interoperability based on the presented taxonomy
3) detailed overview of the open issues and potential future research directions in IoT interoperability.

During the past 6-7 years, there have been several sophisticated survey papers published on IoT [2, 9–13]. They have identified the enabling technologies for actualizing IoT and the different use-cases and applications of IoT. The associated challenges, such as addressing and networking, heterogeneity, context awareness, resource discovery, security and privacy issues have been introduced. In contrast, our survey distinguishes itself from the existing literature by focusing on the essential issues of IoT interoperability, which is fundamental for realizing the vision of a global IoT ecosystem. Two studies partially survey the interoperability challenge [14, 15]. In [16], the authors give a short overview of the challenges of IoT including technical interoperability, semantic interoperability, security and privacy, smart things and resilience and reliability. Further, [17] provides a review of only three IoT interoperability projects (UniversAAL, Domoinstant and AllJoyn) which are limited to the field of Ambient Assisted Living systems and Smart Home environments. However, a comprehensive study dedicated to IoT interoperability is missing in the literature.

This paper provides a comprehensive study on IoT interoperability and presents interoperability definition. Taxonomy of interoperability in IoT is devised from different perspectives to: device interoperability, network interoperability, syntactical interoperability, semantic interoperability, and platform interoperability. Furthermore, based on the provided taxonomy we review the major interoperability handling techniques and solutions used for addressing interoperability. The survey ends by providing some open research challenges. This review helps domain experts and professionals identify the different techniques for improving IoT interoperability to increase the number of interoperable IoT products.

The remainder of this paper is organized as follows. Section 2 introduces the definitions and models of IoT interoperability. In Section 3 a taxonomy for IoT interoperability is provided and in Section 4 we comprehensively survey the interoperability handling approaches in the context of IoT. Finally, we provide an overview of the open issues and potential future research directions in IoT interoperability.

2 IoT interoperability: an overview

The problem of information system interoperability has existed since 1988 [18]; and possibly even earlier. There are several definitions for interoperability in the literature. Among the diverse definitions for interoperability, we quote the ones related to our context. The Oxford Dictionary gives a general definition for interoperability as "*able to operate in conjunction*". This implies that two interoperable systems can understand one another and use the functionality of each other. ISO/IEC defines interoperability as "the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units [19]”. In a broader view, interoperability is defined by IEEE as “*the ability of two or more systems or components to exchange information and to use the information that has been exchanged* [20]”. According to this definition, interoperability is realized by devising standards. In IoT interoperability can be defined as the ability of two systems to communicate and share services with each other [21].

The ability of two systems to interoperate can also be presented using different types of layered models. For example, a six level structure including: *no connection* (no interoperability between systems), *technical* (basic connectivity and network connectivity), *syntactical* (data exchange interoperability), *semantic* (understanding in the meaning of the data), *pragmatic/dynamic* (applicability of the information) and *conceptual* (shared view of the world) is elaborated by Tolk et al. [22]. A similar six level model is proposed in [23] by Pantsar Syvaniemi et al. containing: *connection, communication, semantic, dynamic, behavioural*, and *conceptual*. These six levels are equivalent to the Tolk’s model levels *technical, syntactical, semantic, pragmatic/dynamic* and *conceptual,* respectively.

3 Interoperability in IoT: a taxonomy

To understand interoperability in IoT, we need to take an approach to classifying it. This section of the study describes overview of IoT interoperability taxonomy. The interoperability issues in IoT can be seen from different perspectives due to heterogeneity. Heterogeneity is not a new concept nor restricted to a domain. Even in the physical world there are many types of heterogeneities for example, people speak dissimilar languages, but they can still communicate with each other through a translator (human/tools) or by using a common language. Likewise, the diverse elements comprising IoT (devices, communication, services, applications, etc.) should seamlessly cooperate and communicate with each other to realize the full potential of IoT ecosystem. As indicated in Fig. 1 IoT interoperability can be seen from different perspectives such as device interoperability, networking interoperability, syntactic interoperability, semantic interoperability, and platform interoperability that we examine them as follows.

3.1 Device interoperability

IoT is composed of a variety of devices, even more than the traditional Internet. These devices, which are called “smart objects/things”, may consist of *high-end devices* or *low-end devices* [24]. The *high-end IoT devices* have enough resources and computational capabilities such as Raspberry Pi and smartphones. On the other hand, the *low-end IoT devices* are resource-constrained in terms of energy, processing power and communication capabilities than typical hosts such as RFID tags, tiny and low-cost sensors, and actuators, Arduino, and OpenMote to name a few. The microcontroller (MCU) architecture and key system characteristics of IoT devices such as processor speed, RAM, communication technology, and battery capacity differ broadly between different brands and models Also, various communication protocols have emerged due to the different requirements of IoT markets. For example, IoT devices such as Smart TV, printers, air conditioners support traditional ubiquitous Wi-Fi technologies and 3G/4G cellular communications. Most recent IoT medical devices are based on ANT+ standard; other wearable devices mostly support Bluetooth SMART and NFC, while the environmental sensors use ZigBee-based on IEEE 802.15.4 standard. Besides these protocols, the standard communication protocols are utilised for smart devices, sensor, and actuators (i.e., Z-Wave, ZigBee, and WirelessHart) as well as the non-standard proprietary solution (i.e., LoRa, SIGFOX).

In the absence of a de-facto communication standard(s), not all smart devices implement all these communication technologies. In some cases, the devices that want to exchange information may be using different communication technologies which requires interoperability between the different types of heterogeneous devices that co-exist in the IoT ecosystem. Device interoperability refers to enabling the integration and interoperability of such heterogenous devices with various communication protocols and standards supported by heterogeneous IoT devices. Device interoperability is concerned with (i) the exchange of information between heterogeneous devices and heterogenous communication protocols and (ii) the ability to integrate new devices into any IoT platform.

3.2 Network interoperability

The networks that IoT devices will be operating on will continue to be heterogenous, multi-service, multi-vendor and largely distributed. Different from desktop computers, IoT devices generally rely on various short-ranged wireless communication and networking technologies which is rather more intermittent and unreliable [24]. Network level interoperability deals with mechanisms to enable seamless message exchange between systems through different networks (networks of networks) for end-to-end communication. To make systems interoperable, each system should be able to exchange messages with other systems through various types of networks. Due to the dynamic and heterogenous network environment in IoT, the network interoperability level should handle issues such as addressing, routing, resource optimization, security, QoS, and mobility support [25].

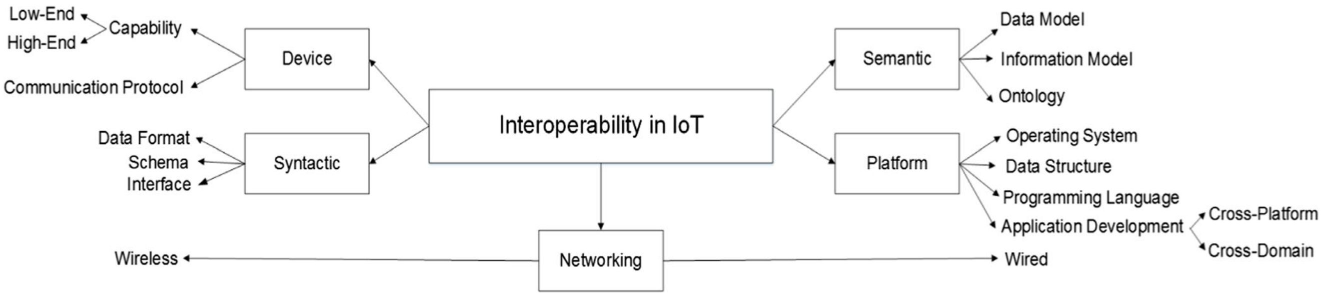

Fig. 1. IoT taxonomy

3.3 Syntactical interoperability

Syntactic interoperability refers to interoperation of the format as well as the data structure used in any exchanged information or service between heterogeneous IoT system entities. An interface needs to be defined for each resource, exposing some structure according to some schema. WSDL and REST APIs are examples. The content of the messages need to be serialized to be sent over the channel and the format to do so (such as XML or JSON). The message sender encodes data in a message using syntactic rules, specified in some grammar. The message receiver decodes the received message using syntactic rules defined in the same or some other grammar. Syntactic interoperability problems arise when the sender's encoding rules are incompatible with the receiver's decoding rules, which leads to mismatching message parse trees.

3.4 Semantic interoperability

The W3C defines semantic interoperability as "enabling different agents, services, and applications to exchange information, data and knowledge in a meaningful way, on and off the Web" [26]. The WoT addresses the current fragmentation by exposing things and systems data and metadata through API. But, such efforts have been hampered because the corresponding parties need to share knowledge of an API [27] and many devices do not speak the same language and cannot exchange across different gateways and smart hubs [28]. To be more precise, the data generated by things about the environment may have a defined data format (e.g. JSON, XML or CSV), but the data models and schemas used by different sources are usually dissimilar and not always compatible. Besides, the data may be represented in diverse units of measurements and consist of other information. This semantic incompatibility between data models and information models results in IoT systems not being able to dynamically and automatically inter-operate as they have different descriptions or understandings of resources and operational procedures, even if IoT systems expose their data and resources to others [27].

3.5 Platform interoperability

Platform interoperability issues in IoT arises due to the availability of diverse operating systems (OSs), programming languages, data structures, architectures and access mechanisms for things and data. There are currently many different OSs developed specifically for IoT devices such as Contiki[8], RIOT[9], TinyOS [29] and OpenWSN [30], each with several versions, to deliver services to users. Besides, the IoT platform providers such as Apple HomeKit, Google Brillo, Amazon AWS IoT, and IBM Watson provide different Oss, programming languages, and data structures. For example, Apple HomeKit supports its own open source language Swift, Google Brillo uses Weave, and Amazon AWS IoT offers SKDs for embedded C and NodeJS. This non-uniformity causes hindrance for application developers to develop cross-platform and cross-domain IoT applications.

Developers need to obtain extensive knowledge of the platform specific APIs and information models of each different platform to be able to adapt their applications from one platform to another. A cross-platform IoT application can access different IoT platforms and integrate data from various platforms. For example, consider the following application scenario: a user who has health problems uses an IoT cross-platform application every day to help him with his everyday tasks. The IoT application connects to the user's smart health platform of wearable sensors to continuously monitor his health conditions (heart rate, fall situation, and glucose level) and in an emergency, locates him and sends an ambulance. The application can also access a smart-city platform to buy a ticket to the users desired destination and shows the fastest route to the bus/train station. The cross-platform interoperability between things and data in this scenario enables interoperability across separate IoT platforms specific to one vertical domain such as smart home, smart healthcare, smart garden, etc. After cross-platform interoperability is enabled, cross-domain interoperability can be achieved in which different platforms within heterogenous domains are federated to build horizontal IoT applications. Fig. 2. shows the concept

[8] www.contiki-os.org

[9] https://riot-os.org

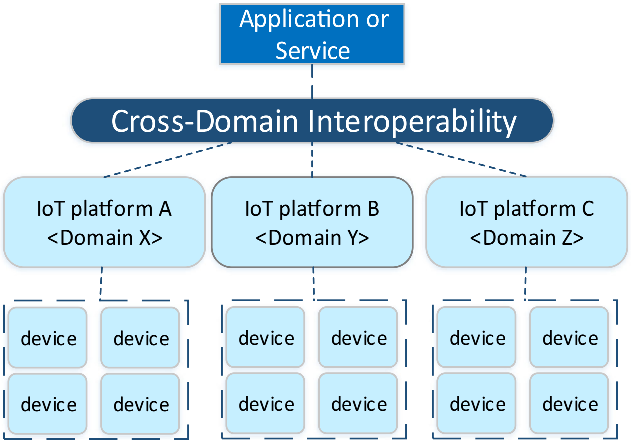
Application or Service
Cross-Domain Interoperability
IoT platform A <Domain X>
IoT platform B <Domain Y>
IoT platform C <Domain Z>
device

Fig. 2. Cross-domain interoperability

behind cross-domain interoperability where different IoT platforms from different IoT domains (e.g. health, home, transport, etc.) can be integrated to build new innovative applications. For example, a smart home platform can provide domain-specific enablers such as air temperature and the lighting conditions.

These enablers can then be exploited by other IoT platforms, such as smart healthcare, to provide more innovative applications and scenarios.

4 Interoperability handling approaches in IoT

To improve the state of IoT interoperability, researchers have leveraged numerous approaches and technologies which we refer to interoperability handling approaches. In the following, we provide an overview of the different interoperability handling approaches for addressing interoperability challenges in IoT. In addition, we provide a summary of a representative sample of proposals for IoT in Table 1. The aim is to provide an overview of the interoperability perspective they focus on and the approaches they take for interoperability. In particular, for each proposal we consider the interoperability perspective (device, network, syntactical, semantic, cross-platform and cross-domain interoperability), interoperability approach, openness, connectivity, application protocols, and security/privacy metrics. The different proposals are divided into IoT standard frameworks, projects, and platforms. We do not cover the recent H2020 projects as they have already been compared in our previous work [9]. Furthermore, the technical details of all the proposals are not included, since the main objective here is to define their interoperability approach.

4.1 Adapters/gateways

Gateways or adapters are the class of schemes which address interoperability through the development of an intermediate tool sometimes called mediators to improve interoperability between IoT devices. The objective here to bridge between different specifications, data, standards, and middleware's etc. To perform a conversion between the protocol of the sending device and the protocol of the receiving device, the gateway can be expanded with the use of plug-ins. For example, when IoT devices use dissimilar communication technologies (i.e., Bluetooth and ZigBee) or when they use dissimilar application layer protocols (i.e., XMPP and MQTT). Gateways can be dedicated hardware, or the function can be embedded in the firmware or software of an intelligent device such as a programmable logic controller (PLC), human-human interface (HMI), or computer. A one-to-one protocol gateway enables interoperability among two types of protocols. This approach has limitation on scalability in terms of the number of different IoT products interacting together requiring specific connectors (design time complexity) and the high number of IoT products in a deployment requiring brokering (runtime complexity). If we suppose to bind n distinct IoT products, the eventual complexity will be n(n-1)/2. Using a single protocol for IoT would impossible. Therefore, several one-to-any protocol gateways are used for providing seamless interoperability.

There are many industrial and academic works which focus on standardization and design of IoT gateways. For example, the Apple HomeKit, Alphabet (Google) Net ecosystem, If-This-Then-That (IFTTT)[10], and Ponte [31] design different connectors to support various IoT device communication protocols. For example, Ponte [31] was initially developed as QEST [32] and is a framework which enables publish and receive of data from sensors and actuators through M2M protocols, accessible through a REST API. It allows the programmer to automatically convert and exchange data between HTTP, CoAP and MQTT. However, the main limitation of Ponte is that it assumes the underlying devices support TCP/IP, and resource-constrained devices have not been taken into account. In addition, Zhu et al. [33] proposes an IoT gateway based on user-space programmable software to bridge the heterogeneity between WSN protocols and mobile communication networks or Internet and includes functionalities like data forwarding, protocol conversion and management. The gateway functionality is realized by a smartphone and connects networks with different protocols such as ZigBee, Bluetooth, GPRS and Ethernet. However, the main limitation of their approach is that users cannot access the sensor data unless they install server software on their PC. The authors of [34] discuss the lack of interoperability in IoT applications and services. The proposed gateway is responsible for the adaption of the different device protocols and for ensuring the proper management and security functionalities. The architecture supports standard and proprietary interfaces which also allows it to extend the gateway capabilities. But, scalability features are

[10] https://ifttt.com

Table 1. Summary of the IoT interoperability proposals, ✓=supported; ✗=not supported; NG = Not Given

	Ref	D	N	Sy	Se	CP	CD	Solution	Openness	Data Format	Application Protocols	Connectivity	Priv/ Sec
Standard Frameworks	oneM2M	✓	✓	✓	✓	✓		Open standard. Gateway, API	✓	-	RESTful HTTP, CoAP, MQTT	Cellular, Zigbee, Bluetooth, WiFi	✓
	OMA LWM2M	✓	✓					Open standard	ISC License	XML	CoAP	Cellular, Zigbee, WiFi	✗
	OGC SWE				✓	✓		sensor data model	GPL License	XML, EXI	RESTful HTTP, MQTT	-	✗
	ETSI Smart M2M	✓	✓	✓				Service layer	✓	XML, JSON, EXI	RESTful HTTP, CoAP	Cellular, Zigbee, Bluetooth, WiFi	✓
	HyperCat			✓				open standard, open API	✓	JSON, RDF	RESTful HTTP	-	✗
	AllJoyn	✓						APIs, Open standard protocols	ISC License	JSON, XML, EXI	Proprietary protocol	WiFi, Bluetooth, NFC, ZigBee	✓
	OIC IoTivity	✓						Industry standard technologies, protocol plug-ins, APIs	Apache License 2.0	XML, JSON	RESTful HTTP, CoAP	WiFi, BLE,	✓
IoT platforms	IFTTT				✓	✓		interoperability as a service	✗	depending on supported services	-	Z-Wave, ZigBee, Bluetooth, WiFi, NFC	✓
	Amazon AWS IoT			✓		✓		Gateway, REST API,	Partially open source (open source libraries)	JSON	HTTP, MQTT, WebSockets	GSM, 3GPP	✓
	OpenRemote	✓	✓					Open APIs	Affero GNU Public License	XML, JSON	HTTPS REST	Z-Wave, KNX, EnOcean, Zigbee, Bluetooth, IFTTT	✓
	ARM mbed	✓						LWM2M	✗	JSON	HTTP, HTPPS, MQTT, CoAP	Ethernet, WiFi, Cellular, 6LoWPAN	✓
	Intel IoT Platform	✓						Open APIs	Intel open source license	XML, JSON	MQTT	ZigBee, Bluetooth, cellular, wifi	✓
	Nimbits			✓				Gateway	Apache License Version 2.0	JSON	HTTP REST, XMPP	NG	NG
	Kaa	✓						Embed SDK into developers chip or device, microservices, edge computing	Apache License Version 2.0	-	MQTT, CoAP, XMPP	WiFi, Ethernet, ZigBee,	✓
	Xively				✓			Open APIs, microservices, platform as a service	Partially open source (open source libraries)	JSON, SenML, XML, CSV, Atom, RSS	HTTP REST, MQTT, CoAP, XMPP, WebSocket	NG	✓
	PTC ThingWorx	✓			✓			Protocol translation, web services (SOAP & REST),	✗	Xml, JSON, CSV, text	HTTP, HTTPs, XMPP, MQTT, WebSockets, DDS, CoAP	WiFi, GSM	✓
	ThingSpeak	✓			✓			Open API, WoT interface	GNU LGPLv3	XML, JSON, CSV	HTTP REST	NG	✓
	WoTkit	✓			✓	✓		Open API, WoT hub	✗	JSON, KML, CSV, HTML	HTTP REST	Bluetooth, ZigBee	✓
	LinkSmart/Hydra	✓	✓	✓				Gateway, web service, edge computing, micro service	LGPLv3	-	MQTT, HTTP REST	Bluetooth, ZigBee, USB	✓
	Node-RED			✓				Open API	Apache license 2.0	JSON	CoAP, MQTT, XMPP	6LowPAN, Thread, ZigBee, Z-wave	✓
IoT projects	Arrowhead			✓				SOA	✓	XML, JSON		NG	✓

Table 1. (continued)

Ref	D	N	Sy	Se	CP	CD	Solution	Openness	Data Format	Application Protocols	Connectivity	Priv/ Sec
										HTTP REST, MQTT, XMPP, CoAP		
Ponte	✓	✓					gateway	Eclipse public license 1.0	XML	RESTul HTTP, CoAP, MQTT	Cellular, ZigBee, Bluetooth, WiFi	✓
SGS				✓			SSN ontology, SemSOS	✗	JSON, RDF, JSON-LD	RESTful HTTP, CoAP, MQTT, XMPP	NG	✗
SOCRADES	✓		✓				SOAP web service, DWPS	✗	XHTML	-	Bluetooth, ZigBee, RFID, 6LowPAN	✓
OpenIoT				✓	✓		Open APIs, SSN,	LGPLv3	XML, JSON, RDF	CoAP, RDF	-	✓
FIWARE				✓	✓		open standard, published APIs	OMA NGSI-9/10 IDAS/IoT Data Edge	XML, JSON	HTTP	-	✓
iCore	✓		✓				Virtualization, semantic web	✗	XML,JSON,JSON-LD	MQTT, CoAP	NG	✓
SpitFire			✓	✓			Semantic web, sensor meta-data	✗	XML, text, RDF	NG	NG	✗
Butler	✓							✗	XML, JSON	CoAP	6LowPAN, Zigbee	✓
UbiROAD				✓	✓	✓		✗	XML	NG	NG	✓
SEG 3.0				✓	✓	✓		✗	RDF(s), SenML, CSV	NG	NG	✗

not discussed. Similarly, efforts like [35, 36] present off-the-shelf smartphones as mobile gateways for IoT interoperability. However, their main limitation is the excessive energy consumption. Asensio et al. propose Common Thing Protocol (CTP) to provide a specification to bring things into the IoT [37] by using an intelligent IoT gateway as a main component in the architecture. The Semantic Gateway as a Service (SGS) is presented as a gateway between the physical world and the high-level layers of an IoT system. According to the SGS architecture, raw sensor data are transferred from external sink nodes to the central gateway node via the multi-protocol proxy. Before being forwarded, data are semantically annotated using W3C SSN ontology, SemSOS tool and other domain specific ontologies. Semantic annotation of sensor data provides semantic interoperability between messages and supply higher-level actionable knowledge for implementing.

4.2 Virtual networks/ overlay-based solutions

Virtual networks or Overlay-based solutions have been proposed in [38] the “Managed Ecosystems of Networked Objects” (MENO), with the aim to integrate sensor and actuators and other IP-smart objects seamlessly to the Internet for end-to-end communication. The main idea behind MENO is to create a virtual network on top of physical networks and thereby allow communication with other types of devices, including sensor nodes. Within each virtual network, end-to-end communication is possible using different protocols. Once end-to-end communication is enabled, it becomes possible for application developers to write new applications that utilize sensors, actuators, and other devices. It appears to be on track to use a clean-slate approach to integrate the physical work with the Internet in a seamless way. The concept utilized by MENO is used to develop the Internet of Things Virtual Network (IoT-VN) [39] shown in Fig. 3 to integrate smart-resource constrained devices into the Internet. This is achieved by creating a virtual network of all the devices that want to communicate and cooperate. Their solution focuses on both resource-constrained and non-constrained things. This integration is achieved by integrating all involved devices into a secured virtual network, named an Internet of Things Virtual Network (IoT-VN). The advantage of this approach is enabling end-to-end communication between devices, however the key issues are scalability and binding to specific protocols.

4.3 Networking technologies

Different networking protocols and technologies have been used to provide networking interoperability in IoT. For example, the conventional Universal Plug and Play (UPnP) and DLNA protocols is used for communication between IoT devices and the gateway. In the following, we discuss the

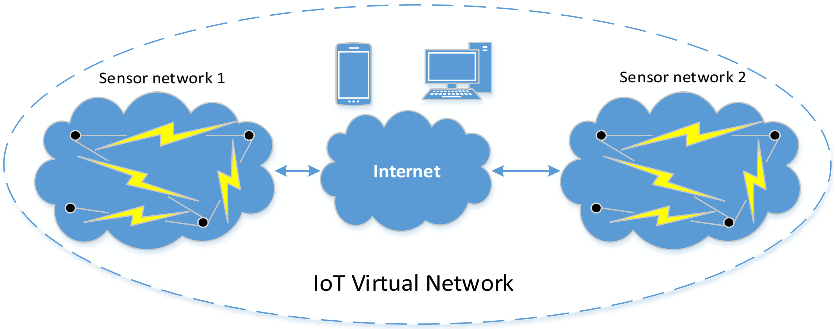

Fig. 3. Virtual network

main technologies/solutions for interoperability at the network level.

4.3.1 IP-based approaches

The IP-based approaches embed the full TCP/IP stack on smart devices. By embedding the TCP/IP stack in Fig. 4, the sensor and actuators are directly connected to the IP network to allow end-to-end communication between sensor network and IP network. Therefore, the sensor and actuators are directly connected to the IP network to allow end-to-end communication between sensor network and IP network. Some have attempted to implement the TCP/IP stack on sensor nodes such as uIP [40], TinyTCP [41], and lwIP [42]. The key benefit of implementing the TCP/IP stack on sensor nodes is that gateways and protocol translations are not required. However, the authors of [43] argue that an all IP sensor network is not possible on sensor nodes because of their resource-constraint property. Due to the success of these implementation, the IETF has formed working groups (WGs) at the network layer such as Routing Over Low Power and Lossy Networks (ROLL) [44], IPv6 over Low Power WPAN (6LoWPAN), Constrained Application Protocol (CoAP) which is based on UDP, and Constrained Restful Environment to solve the connectivity problem of resource-constrained devices. This approach, still uses gateways to convert between standard protocols used in the Internet and proprietary protocols used in the sensor network, e.g. IPv6 to 6LoWPAN. Therefore, due to the use of standard protocols, this approach does not have the limitations of the gateway-based approaches. The key benefit is that the gateway and the sensor nodes do not have to be from the same vendor which improves the interoperability between devices. IP as the de facto standard of the Internet provides a single open standard interface for a trillion things. However, by permitting direct access with the resource-constrained devices, security related issues like authentication and access control are presented. The security challenges in the IP-based approaches are detailed in [45].

4.3.2 Software-defined networking (SDN)

Software defined networking (SDN) [46] is a new networking paradigm to make the current wireless and mobile networks more “intelligent”, efficient, secure, and scalable in order to handle the large amount of data produced in the IoT [47]. One of the main novelties of SDN for breaking the vertical silos in IoT, is to separate the control and data planes in networking devices. Fig. 5 illustrates a simplified view of the integration of IoT and SDN.

SDN has been applied to IoT to facilitate networking applications such as heterogeneity [48, 49], mobility management [50, 51], QoS management [52, 53], and security [54]. For instance, Martinez-Julia and Skarmeta [48] used SDN to allow different objects from different networks to communicate with each other using IPv6 and at the same time simplify the management and control operations of various objects types by adding an additional IoT controller over the SDN controller. Thus, even so the devices have different protocols, the forwarding devices in the router convert it in a form understandable by the receiver. This enables the communication of diverse devices in the network. Another work that emphasises the necessity to deal with the heterogeneity of the diverse

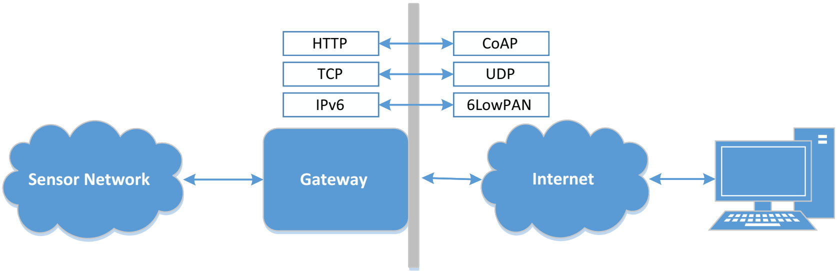

Fig. 4. IP-based approaches

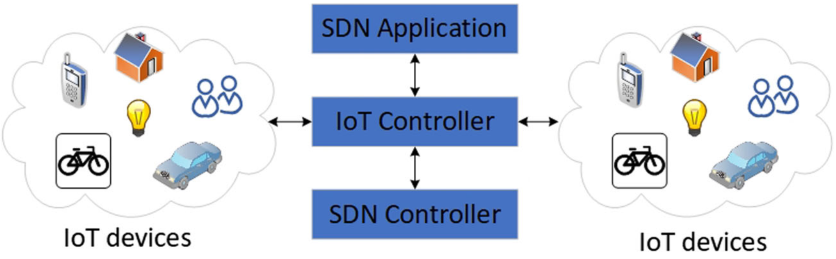

Fig. 5. Integration of IoT and SDN

IoT devices and applications is presented in [49]. The authors conclude that, using the IPv6 may be a suitable choice to handle the large number of connected devices, but the heterogeneity in terms of the diverse characteristics and capabilities is still an open research issue. To address it, they provide a rather high-level architecture of an IoT controller, which to a generic level seems an adequate framework to handle heterogenous IoT flows.

In [50], the authors proposed a new mobility service adapted for sSDN concept to solve the performance issues of PMIPv6 protocol. The authors argue that their solution can be used for mobility management instead of PMIPv6 without using the legacy IPv4 protocol. A middleware is designed and implemented by Qin, Z. et al. [52], which is composed of a layered IoT SDN controller to manage distributed, heterogeneous, and dynamic IoT multinetwork. In their research, a central controller monitors the existing resources and schedules the data streaming according to the specific service requirement e.g., a minimum data rate, maximum tolerable delay or packet loss for each separate flow. IoT SDN exploits network calculus to model the end-to-end flow performance in IoT multi-network environments, semantic modelling for resource matching and the genetic algorithm schedules flows, to optimize the usage of the existing IoT network opportunities. The performance results show that the genetic algorithm based flow scheduling algorithm has better performance compared to bin packing and load balance algorithms.

4.3.3 Network function virtualization

A complementary approach to SDN is network function virtualization (NFV). NFV separates the physical network equipment's (i.e., network address translator, firewall) from the functions that run on them. This way, numerous service providers can create several isolated virtual networks which could then share the physical network equipment's provided by the network infrastructure providers. NFV has the potential to reduce Operational Expenditure (OPEX) and Capital Expenditure (CAPEX) costs by sharing the network infrastructure, dynamic scaling, on-the-fly, and flexible network function deployment [55].

An example where NFV is used in IoT is [56], they defined their own abstract IoT architecture which is then combined with SDN architecture (Application, Control and Infrastructure layers) to produce a general SDN-IoT framework. This consists of an upper layer with servers providing developers with the necessary APIs for IoT applications, a middle layer, which contains a distributed network OS, commanding several physically distributed SDN controllers, a south layer, which contains the SDN-enabled network switches, and the IoT gateway, which connects them to the middle layer. In essence, this is just the classic SDN architecture, with IoT applications in mind. The authors take it one step further when they claim that, to achieve an IoT-optimized network, one must design the network OS, which sits in the middle layer, using virtualization techniques. The network OS must be used in such a way that the diversity of use-cases and IoT devices is acknowledged. The exact details of using virtualization in the middle layer is missing, but linking NFV techniques with an SDN orchestration logic for an IoT network is noteworthy.

4.3.4 Fog computing

The cloud has been used as a medium to address interoperability called the Fog of Things [57], where the computing, storage and networking services are placed at the edge of the network rather than centralized cloud servers, i.e., as close as possible to the end user devices. This decreases network latency that arises when converting the raw data produced by resource-constrained mobile devices and sensors into knowledge or actionable instructions. Fog computing paradigm provides value to the data before making it available to the web facilitating interoperability in IoT, 5G, AI, tactile internet, virtual reality, and other complex data and network intensive applications [58] and preparing the managed data for further applications to be interoperable [59]. Fog computing provides interoperability of local ecosystems in the fog and also at the cloud level.

4.4 Open API

API is an interface provided by service providers that exposes data or functions to an application written in a high-level language. Publicly available APIs, for providing cross-platform and cross-domain interoperability focuses on well-documented open APIs that provides developers with streamlined access to functionalities and services. There are many popular APIs such Google Maps, YouTube, Flickr, Twitter, Amazon, and Facebook. Today's IoT platforms almost all provide a public API to assist developers access their services. The APIs are usually based on RESTful principles, and allow common operations such as PUT, GET, PUSH, or DELETE. Only three of the studied IoT platforms did not include a REST API for easing the development of web services (i.e. LinkSmart[11], IFTTT and OpenIoT[12]), but use different interaction means. However, the majority of IoT platform providers develop and deploy APIs that are platform-specific and proprietary relying on internal information models to define the syntax of specific operations to be used by their consumers. For example, a mobile application may offer to control your Internet-connected refrigerator. It may have functionalities like showing the items inside the refrigerator, notify you with the expiry date of the ingredients, or start/stop an operation. Without a standard API, if the mobile application wants to integrate more than one refrigerator vendor, it must write custom code to use another platform-specific API, which is a substantial burden for the application developers. However, a standard API enables cross-platform interoperability between the existing solutions with minimal change in the application.

With the massive development of IoT platform providers a vast silo of diverse APIs has been created that increases the difficulty of developing applications as well as interoperability issues. To overcome the effect of API heterogeneity in IoT, some platforms such as ThingSpeak[13] enable the creation of widgets written in Javascript, HTML and CSS that may be distributed on the platform to other users. HyperCat[14] is a specification which provides syntactic interoperability between different APIs and services based on a Catalog that can be tagged with metadata. The catalog contains many resources identified by its URI. Moreover, the symbIoTe[15] and Big-IoT[16] European projects are working on a generic interworking API to provide uniform access to resources of all existing and future IoT platforms to address syntactic and cross-platform interoperability. The Interworking API acts like an adapter which needs to be implemented by other platforms.

4.5 Service oriented architecture (SOA)

To provide syntactic interoperability between heterogeneous devices and across all systems, researchers have proposed Service Oriented Architecture (SOA) as a major technology in different ways [60–63]. SOA is built on top of the network layer so that data and information processing can be easily managed through different service components [64, 65]. In the SOA of the IoT, the interaction with and operations of different wireless devices are classified into different service components and the application layer software can access resources exposed by devices as services. Exposing each component's functionalities as a standard service can significantly increase the interoperability of both network and device. In particular, the Web Service technology has been proposed for realizing the SOA promise of maximum service sharing, reuse, and interoperability [66]. The classic web service oriented approach (WS-* web service) [61, 67] and resource oriented approach (REST web services) [68, 69] have been used to address syntactic interoperability. A study conducted by Pautasso et al [70] compared REST web services with WS-* servers and they concluded that RESTful services are preferred for tactical, ad-hoc integration over the Web, while WS-* are preferred for professional enterprise application integration scenarios.

An extension to SOA named Event-driven SoA (EDSOA) [71] has been proposed for constructing IoT services. Event-driven architecture (EDA) is integrated with SOA to compose IoT services. SOA breaks the application into multiple independent services described by the standard interface specification, whereas EDA coordinates independent services using event flows. The authors focus on building a scalable EDSOA which could use resource information to compose IoT services, use independent and shared events to run those services, and then use event sessions to coordinate the services.

4.6 Semantic web technologies

Originally, the Semantic Web technologies developed by the W3C such as Resource Description Framework (RDF), SPARQL and Web Ontology Language (OWL) have been used for describing resources on the Web. Currently, the same standards are used in many different areas including IoT. The Semantic Web of Things (SWoT) [72] paradigm is proposed for the integration of the Semantic Web with the WoT, for realizing a common understanding of the various entities which form the IoT. Recent research has concluded that semantic web technologies are a major driver for interoperability across heterogenous environments [73]. The literature uses semantic web technologies to achieve semantic interoperability by using standards or agreements on the format and meaning of data or in a dynamic way by using shared vocabularies either in a schema form and/or in an ontology-driven approach.

Ontologies (or vocabularies) in IoT are a set of objects and relationships used to define and represent an area of concern. They represent an abstraction technology which aims to hide heterogeneity of IoT entities, acting as a mediator between IoT

[11] https://docs.linksmart.eu
[12] http://www.openiot.eu
[13] http://thingspeak.com
[14] www.hypercat.io
[15] http://iot-epi.eu/project/symbiote
[16] http://big-iot.eu

application provider and consumers, and to support their semantic matchmaking [74]. Many ontologies have been proposed in the context of IoT such as W3C Semantic Sensor Network (SSN) [44], IoT-Ontology, SAREF and OpenIoT. A comprehensive survey of the existing ontologies which are ready to be used in three different domains: general IoT ontologies, health, and transportation and logistics can be found in [75]. They also outline an approach using ontologies to achieve semantic interoperability among heterogeneous IoT platforms. The authors believe that the SSN ontology has seen the strongest adoption and inspired other projects. However, no single domain has a global ontological standard, and most application specific ontologies are proprietary.

There are several IoT research projects which utilize the capabilities of the above-mentioned ontologies or other semantic technologies to improve semantic interoperability such as Semantic Sensor Web (SSW) [76], OpenIoT, HYDRA[17], SPITFIRE [77], SENSEI[18] to name a few. The SSW is one of the initial studies on semantic IoT/WoT concept, usually understood as a marriage of Sensor Web and Semantic Web technologies. The Open Geospatial Consortium (OGC) has developed SensorML[19] which is only a syntactic standard for sensor web enablement (SWE) using XML-based protocols and APIs without providing however, either semantic interoperability nor a basis for reasoning. UbiROAD [78] achieves semantic interoperability by two layers: 1) data-level interoperability and 2) functional protocol-level interoperability and coordination. Serrano [79] discuss the semantic interoperability challenges in the context of IoT and present SEG 3.0 methodology to provide semantic interoperability between heterogenous applications. The methodology uses semantic web technologies to combine heterogeneous IoT data, as well as adding value to the data to assist developers and IoT practitioners for building IoT applications. The framework consists of 12 layers which focus on heterogeneity of devices, communication networks, data, reasoning and services. The authors of [80] present the idea of "sensing as a service", where standard service technologies are used as an interface that represents the IoT resources (i.e. the physical world devices) and provide an access to the functions and capabilities of these resources. In this work a set of semantic models for IoT resources, entities and services is presented. These semantic models for the IoT component descriptions offers interoperability at the data and service layers.

4.7 Open standard

Open standards are one significant means to provide interoperability between and within different domains. A standard is framework of specification that has been approved by a recognized organization, or is generally accepted and widely used throughout by the industry [81]. Currently there are several standard bodies, consortiums and alliances trying to solve IoT standard issues including Open Interconnect Consortium (OIC) providing IoTivity[20], AllSeen Alliance providing AllJoyn, oneM2M[21], OMA LWM2M[22] and ETSI M2M[23]. The IPSO alliance focuses on semantic interoperability in IoT and the standardization of resource-based object model which is based on standards like SenML, CoAP and 6LoWPAN. Frameworks such as LWM2M and IoTvitiy work with the IPSO alliance. The IoTivity focuses on device interoperability irrespective of form factor, operating system or service provider through protocol plug-ins. The AllJoyn framework functions as a software bus between devices facilitating device interoperability for home automation and industrial lighting applications. Constrained devices use a thin library, and do not have a bus attachment. This framework introduces high overhead for low end devices. The framework has also an open source codebase and various modular services which ensures interoperability. OneM2M enables interoperability on the platform level using a horizontal service layer for M2M and IoT communications, that is network independent and offers internetworking to different existing M2M vertical systems. Syntactic and semantic interoperability between platforms are achieved by using ontologies.

5 Open challenges

Although the IoT standards, platforms and projects presented in this work help advancing IoT interoperability issues, there are still some open research challenges to be solved which is the case for any new paradigm. This survey shows that there have been important developments in the area of IoT interoperability, with the subsequent research challenges remaining.

- Most of the surveyed IoT proposals focus on interoperability from a specific perspective rather than providing interoperability among all the mentioned perspectives. In particular, it is clear from Table 1 that cross-domain interoperability support is limited and is not considered in most proposals, except oneM2M, UbiROAD and SEG 3.0 (Table 1). Rather, the solutions tend to focus more on the lower levels like the device and the network layers. There is evidently substantial room for future work in this area.

[17] www.hydramiddleware.eu
[18] www.sensei-iot.org
[19] www.ogcnetwork.net/sensorml

[20] https://www.iotivity.org
[21] www.onem2m.org
[22] http://technical.openmobilealliance.org/Technical/technical-information/omna/lightweight-m2m-lwm2m-object-registry
[23] www.etsi.org

Using semantic web technologies and interworking API could be a good starting point for providing cross-platform interoperability.

- IoT devices have a key role in realizing the IoT, thus it is vital to consider their capabilities in addressing interoperability. An ideal IoT platform would offer a pool of standardized communication protocols where the device manufacturers may select the appropriate protocols (e.g. CoAP for constrained devices). However, in the absence of a defacto communication standard(s), not all smart devices implement all these communication technologies. It is crucial that a standardized protocol is established for all devices like the existing efforts performed by IETF and ETSI M2M for low end devices. Therefore, a realistic interoperability solution should not rely on a network entity like a gateway. Since the gateway solutions have limitations when changes occur (a new device is added or upon updates). Furthermore, device to device communication (D2D) requires a gateway free interoperability solution to be scalable.
- Even the most popular IoT platforms do not consider edge computing paradigms for speed and efficiency expect for Kaa[24], LinkSmart, and ThingWorx[25] (Table 1).
- Today's IoT platforms almost all provide a public REST API to access the services, only three of the studied IoT platforms did not include a REST API i.e. LinkSmart, IFTTT and OpenIoT (Table 1). These APIs are generally compliant with the RESTful principles; however, most platforms use custom REST APIs and data models which makes mashing up of data across multiple platforms difficult.
- To enable an IoT ecosystem the interoperability frameworks should consider connecting more than two platforms together. The solutions should be realistic and scalable to multiple platform with the possibility to add additional platforms when new platforms appear. The current solutions do not scale to a group of IoT platforms and only consider specific scenarios.
- Enabling interoperability between different platforms implies that different platforms which have been previously deployed with different technologies (even non-IoT) and underlying features and (probably) belong to different vendors to be integrated. The interoperability should be made possible irrespective of the underlying technologies.
- Providing interoperability between IoT platforms should not require the stakeholders to adapt to major changes in their systems, and the solution should not be dependent on their system.
- There are currently several different academia, industry, and standardization bodies aiming to solve IoT system interoperability. It is not likely that a common set of standards will be universally accepted which will allow IoT devices and platforms to work together.
- Interoperability testing of solutions and standards to solve the different types of interoperability is still a challenge. Currently the process of testing the effectiveness of a solution involves different stakeholders (vendors, developers and service providers) to participate to face-to-face meetings, i.e. plugtests, to validate their implementation against existing standards. This process involves extensive testing and is labour-intensive. Thus, interoperability testing needs to be automated to inspire small business to develop interoperable solutions.

6 Conclusion

Improving interoperability in IoT is fundamental for the success of IoT. Since the emergence of IoT many different proposals have focused on this crucial issue. The proposals are diverse and promote different approaches. This article takes these works into account and presents a comprehensive overview of the topic. By doing this, the taxonomy of IoT interoperability was identified. Furthermore, we studied and classified the related strategies for handling specific types of interoperability. According to the different interoperability types and interoperability handling approaches, a comprehensive survey on the recent state-of-the-art research has been presented. Finally, open research issues, challenges and recommended possible future research directions are outlined.

This survey categorized the existing proposals according to their interoperability handling techniques: gateways, virtual network, networking technologies, open API, SOA, semantic web technologies and open standards. Each category has many interoperability proposal, the most significant ones have been presented in this work. Obviously, it is not possible to analyse all related IoT proposal and platforms. Most of the proposals have been summarized (Table 1). The summaries show that the majority of the proposals support at least two of the interoperability types. Semantic interoperability support is limited. Only seven out of the 30 reviewed IoT proposals provide semantic descriptions of their data or services.

Although there are several academic and industry proposals to address IoT interoperability issues, still there is no appropriate ground that can cover some related research issues. The lack of standards and absence of cutting-edge technologies slows the development of IoT. Providing semantically interoperable platforms across the different IoT domains has a clear requirement for research improvements. We believe there is still significant room for future work on this topic.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Ashton K (2009) The internet of things. *RFiD J.* 22(7):97–114
2. Atzori L, Iera A, Morabito G (2010) The Internet of Things: A survey. *Comput. Networks* 54(15):2787–2805
3. Gubbi J, Buyya R, Marusic S, Palaniswami M (2013) Internet of Things (IoT): A vision, architectural elements, and future directions. *Futur. Gener. Comput. Syst.* 29(7):1645–1660
4. van Kranenburg R (2008) The internet of things: a critique of ambient technology and the all-seeing network of RFID. Inst Netw Cult 2
5. Sundmaeker H, Guillemin P, Friess P, Woelfflé S (2010) Vision and challenges for realising the Internet of Things. *Clust. Eur. Res. Proj. Internet Things, Eur. Commision* 3(3):34–36
6. Unify-IoT project, “Deliverable D03.01 Report on IoT platform activities - UNIFY-IoT,” 2016
7. Manyika J, Chui M, Bisson P, Woetzel J, Dobbs R, Bughin J, Aharon D (2015) The internet of things: mapping the value beyond the hype. McKinsey global institute. McKinsey Glob Inst 3
8. Macaulay T (2016) RIoT control: understanding and managing risks and the internet of things. Morgan Kaufmann
9. Mahda MN, Mohammed A, Gaedke M (2017) Interoperability in internet of things infrastructure: classification, challenges, and future work (In Press)
10. Perera C, Zaslavsky A, Christen P, Georgakopoulos D (2014) Context Aware Computing for The Internet of Things: A Survey. *IEEE Commun. Surv. & Tutorials* 16(1):414–454
11. Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M (2015) Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun. Surv. & Tutorials Tutorials* 17(4):2347–2376
12. Da Xu L, He W, Li S (2014) Internet of things in industries: A survey. *IEEE Trans. Ind. informatics* 10(4):2233–2243
13. Bandyopadhyay S, Sengupta M, Maiti S, Dutta S (2011) A survey of middleware for Internet of things. Commun Comput Inf Sci vol 162 CCIS, pp 288–296
14. Gazis V, Goertz M, Huber M, Leonardi A, Mathioudakis K, Wiesmaier A, Zeiger F (2015) Short paper: IoT: challenges, projects, architectures. pp 145–147
15. Gambi E, Montanini L, Raffaeli L, Spinsante S (2016) Interoperability in IoT infrastructures for enhanced living environments
16. Gazis V, Goertz M, Huber M, Leonardi A, Mathioudakis K, Wiesmaier A, Zeiger F (2015) Short paper: IoT challenges, projects, architectures. In Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on, pp 145–147
17. Gambi E, Montanini L, Raffaeli L, Spinsante S (2016) Interoperability in IoT infrastructures for enhanced living environments. In Interoperability in iot infrastructures for enhanced living environments, pp 1–5
18. H. yliopisto. Department of Computer Science, F. Eliassen, and J. Veijalainen, A functional approach to information system interoperability. 1988
19. “ISO/IEC 2382-1:1993 Information Technology – Vocabulary – Part 1: Fundamental terms. International Organization for Standardization (ISO).” [Online]. Available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=7229
20. Radatz J, Geraci A, Katki F (1990) IEEE standard glossary of software engineering terminology. *IEEE Std* 610121990(121990):3
21. Kiljander J, D’Elia A, Morandi F, Hyttinen P, Takalo-Mattila J, Ylisaukko-Oja A, Soininen JP, Cinotti TS (2014) Semantic interoperability architecture for pervasive computing and internet of things. *IEEE Access* 2:856–873
22. Tolk A (2004) Composable mission spaces and M&S repositories–applicability of open standards. In Spring simulation interoperability workshop, Arlington (VA)
23. Pantsar-Syväniemi S, Purhonen A, Ovaska E, Kuusijärvi J, Evesti A (2012) Situation-based and self-adaptive applications for the smart environment. *J. Ambient Intell. Smart Environ.* 4(6):491–516
24. Hahm O, Baccelli E, Petersen H, Tsiftes N (2016) Operating Systems for Low-End Devices in the Internet of Things: A Survey. *IEEE Internet Things J.* 3(5):720–734
25. Bello O, Zeadally S, Badra M (2016) Network layer inter-operation of Device-to-Device communication technologies in Internet of Things (IoT). *Ad Hoc Networks* 0:1–11
26. W3C, “W3C Semantic Integration & Interoperability Using RDF and OWL.” [Online]. Available: https://www.w3.org/2001/sw/BestPractices/OEP/SemInt/. [Accessed: 25-Jul-2017]
27. Bauer M, Davies J, Girod-genet M, Underwood M (2016) Semantic interoperability for the web of things
28. Shiao M (2015) Internet of things standardisation and architectures - workshop report
29. E. and others Levis, Philip and Madden, Sam and Polastre, Joseph and Szewczyk, Robert and Whitehouse, Kamin and Woo, Alec and Gay, David and Hill, Jason and Welsh, Matt and Brewer, “TinyOS: An operating system for sensor networks,” Ambient Intell, vol 35, pp 115–148, 2005
30. Thomas KW, Vilajosana X, Kerkez B, Chraim F, Weekly K, Wang Q, Glaser S, Pister (2012) OpenWSN: a standards-based low-power wireless development environment. *Trans. Emerg. Telecommun. Technol.* 23(5):480–493
31. “Ponte - M2M Bridge Framework for REST developers.” [Online]. Available: http://www.eclipse.org/proposals/technology.ponte/. [Accessed: 24-Oct-2016]
32. Collina M, Corazza GE, Vanelli-Coralli A (2012) Introducing the QEST broker: scaling the IoT by bridging MQTT and REST. IEEE Int Symp Pers Indoor Mob Radio Commun PIMRC pp 36–41
33. Zhu Q, Wang R, Chen Q, Liu Y, Qin W (2010) IOT gateway: bridgingwireless sensor networks into internet of things. 2010 IEEE/IFIP Int Conf Embed Ubiquitous Comput pp 347–352
34. Fantacci R, Pecorella T, Viti R, Carlini C (2014) Short paper: overcoming IoT fragmentation through standard gateway architecture. 2014 IEEE World Forum Internet Things, WF-IoT 2014, pp 181–182
35. Pereira C, Rocha P, Santiago F, Sousa J (2016) IoT interoperability for actuating applications through standardised M2M communications. In World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2016 IEEE 17th International Symposium on A, pp 1–6
36. Aloi G, Caliciuri G, Fortino G, Gravina R, Pace P, Russo W, Savaglio C (2016) Enabling IoT interoperability through opportunistic smartphone-based mobile gateways. J Netw Comput Appl no July pp 1–11
37. Asensio Á, Marco Á, Blasco R, Casas R (2014) Protocol and Architecture to Bring Things into Internet of Things. *Int. J. Distrib. Sens. Networks* 10(4):158252
38. Hoebeke J, De Poorter E, Bouckaert S, Moerman I, Demeester P (2011) Managed ecosystems of networked objects. *Wirel. Pers. Commun.* 58(1):125–143

39. Ishaq I, Hoebeke J, Moerman I, Demeester P (2012) Internet of things virtual networks: bringing network virtualization to constrained devices. 2012 IEEE Int Conf Cyber Phys Soc Comput
40. "uIP TCP/IP stack." [Online]. Available: http://users.ece.utexas.edu/~mcdermot/arch/projects_fall_09/Team_04/project/uip-1.0/doc/html/main.html
41. Han G, Ma M (2007) Connecting sensor networks with IP using a configurable tiny TCP/IP protocol stack. In Information, Communications & Signal Processing, 2007 6th International Conference on, pp 1–5
42. Dunkels A (2001) Design and Implementation of the lwIP TCP/IP Stack. *Swedish Inst. Comput. Sci.* 2:77
43. Zuniga M, Krishnamachari B (2003) Integrating future large-scale wireless sensor networks with the internet. USC Comput Sci Tech Rep
44. Thubert P (2012) Objective function zero for the routing protocol for low-power and lossy networks (RPL)
45. Chasaki D, Mansour C (2015) Security challenges in the internet of things. *Int. J. Space-Based Situated Comput.* 5(3):141
46. Kreutz D, Ramos F (2015) Software-Defined Networking: A Comprehensive Survey. *Proc. IEEE* 103(1):14–76
47. Bizanis N, Kuipers F (2016) SDN and virtualization solutions for the Internet of Things: A survey. *IEEE Access* 99:5591–5606
48. Julia PM, Skarmeta AF (2014) Extending the internet of things to IPv6 with software defined networking. white Pap
49. Jararweh Y, Al-Ayyoub M, Darabseh A, Benkhelifa E, Vouk M, Rindos A (2015) SDIoT: a software defined based internet of things framework. *J. Ambient Intell. Humaniz. Comput.* 6(4):453–461
50. Tantayakul K (2016) Impact of SDN on mobility management. In Advanced Information Networking and Applications (AINA), 2016 IEEE 30th International Conference on, pp 260–265
51. Nguyen T, Bonnet C (2016) SDN-based distributed mobility management for 5G networks. In Wireless Communications and Networking Conference (WCNC), 2016 IEEE, pp 1–7
52. Qin Z, Denker G, Giannelli C, Bellavista P, Venkatasubramanian N (2014) A software defined networking architecture for the internet-of-things. In Network Operations and Management Symposium (NOMS), 2014 IEEE, pp 1–9
53. Systems C, France SA, Thubert P, Palattella MR, Engel T (2015) 6TiSCH centralized scheduling: when SDN meet IoT. In Standards for Communications and Networking (CSCN), 2015 IEEE Conference on, pp 42–47
54. Flauzac O, Alez CG (2015) SDN based architecture for IoT and improvement of the security
55. Mijumbi R, Serrat J, Gorricho JL, Bouten N, De Turck F, Boutaba R (2016) Network function virtualization: State-of-the-art and research challenges. *IEEE Commun. Surv. & Tutorials Tutorials* 18(1):236–262
56. Li J, Altman E, Touati C (2015) A General SDN-based IoT Framework with NVF Implementation. *ZTE Commun.* 13(3):42–45
57. Prazeres M, C'assio, Serrano (2016) SOFT-IoT: self-organizing FOG of things. In Advanced Information Networking and Applications Workshops (WAINA), 2016 30th International Conference on, pp 803–808
58. Ai Y, Peng M, Zhang K (2017) Edge cloud computing technologies for internet of things: A primer. Digit Commun Networks
59. Gyrard A, Serrano M, Patel P (2017) Building interoperable and cross-domain semantic web of things applications. Manag Web Things pp 305–324
60. Erl T (2005) Service-oriented architecture (SOA): concepts, technology, and design. Prentice Hall
61. Guinard D, Trifa V, Karnouskos S, Spiess P, Savio D (2010) Interacting with the SOA-based internet of things: Discovery, query, selection, and on-demand provisioning of web services. *IEEE Trans. Serv. Comput.* 3(3):223–235
62. Miorandi D, Sicari S, De Pellegrini F, Chlamtac I (2012) Internet of things: Vision, applications and research challenges. *Ad Hoc Networks* 10(7):1497–1516
63. Li S, Da Xu L, Zhao S (2015) The internet of things: a survey. *Inf. Syst. Front.* 17(2):243–259
64. Vinoski S (2003) Integration with Web Services. *IEEE Internet Comput.* 7(6):75–77
65. Li S, Oikonomou G, Tryfonas T, Chen TM, Da Xu L (2014) A distributed consensus algorithm for decision making in service-oriented internet of things. 10 2 pp 1461–1468
66. Den Heuvel V, Van Den Heuvel MPPW (2007) Service oriented architectures: approaches, technologies and research issues
67. Alam S, Noll J (2010) A semantic enhanced service proxy framework for internet of things
68. Varga P, Blomstedt F, Ferreira LL, Eliasson J, Johansson M, Delsing J, de Soria IM (2016) Making system of systems interoperable - the core components of the arrowhead framework. J Netw Comput Appl no August
69. Vega-barbas M, Casado-mansiua D, Valero MA, Lpez-de-ipina D, Bravo J, Florez F (2012) Smart spaces and smart objects interoperability architecture (S3OiA) CPS. In Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on, pp 725–730
70. Pautasso C, Zimmermann O, Leymann F (2008) Restful web services vs. big'web services: making the right architectural decision. In Proceedings of the 17th international conference on World Wide Web, pp 805–814
71. Zhang JL, Yang, Duan, Li, Chen (2014) Event-driven soa for iot services. In Services Computing (SCC), 2014 IEEE International Conference on 2(2) pp 629–636
72. Scioscia F, Ruta M (2009) Building a semantic web of things: issues and perspectives in information compression. ICSC 2009 - 2009 IEEE Int Conf Semant Comput pp 589–594
73. Jara AF, Antonio J, Olivieri AC, Bocchi Y, Jung M, Kastner W, Skarmeta (2014) Semantic Web of Things: an analysis of the application semantics for the IoT moving towards the IoT convergence. *International Journal of Web and Grid Services* 10:244–272
74. Sheng M, Qin Y, Yao L, Benatallah B (2017) Managing the web of things: linking the real world to the web. Morgan Kaufmann
75. Ganzha M, Paprzycki M, Pawłowski W, Szmeja P, Wasielewska K (2016) Semantic interoperability in the internet of things: an overview from the INTER-IoT perspective. J Netw Comput Appl
76. Sheth SS, Amit, Henson, Cory, Sahoo (2008) Semantic sensor web. IEEE Internet Comput, vol 12(4)
77. M. and others Pfisterer, Dennis and Romer, Kay and Bimschas, Daniel and Kleine, Oliver and Mietz, Richard and Truong, Cuong and Hasemann, Henning and Kr{"o}ller, Alexander and Pagel, Max and Hauswirth (2011) SPITFIRE: towards a semantic web of things. IEEE Commun Mag vol 49 no. 11, pp 40–48
78. Terziyan D, Vagan, Kaykova, Olena, Zhovtobryukh (2010) UbiRoad: semantic middleware for context-aware smart road environments. In Internet and web applications and services (iciw), 2010 fifth international conference on, vol 35 pp 295–302
79. Gyrard A, Serrano M (2016) Connected smart cities: interoperability with SEG 3.0 for the internet of things. In Advanced Information Networking and Applications Workshops (WAINA), 2016 30th International Conference on, no 2 pp 796–802
80. Bauer M, Martinbauerneclabeu E, Meissner S (2011) Service modelling for the internet of things. In Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on, pp 949–955
81. Almeida F, Oliveira J, Cruz J (2011) Open standards and open source: enabling interoperability. *Int. J. Softw. Eng. Appl.* 2(1):1–11