STM JOURNALS

International Journal of
Radio Frequency Innovations

Volume 1, Issue 1, 2023
Jan- June
DOI (Journal): 10.37591/IJRFI

https://journals.stmjournals.com/ijrfi

Review IJRFI

An Interoperability on the Internet of Things (IoT): A Review

Chaitali A. Chate[1,*], Sanjeev S. Sannakki[2], Vijay S. Rajpurohit[3]

Abstract

With technological advancement growing at a rapid rate, we are presented with numerous ways of connecting ourselves to the rest of the world. The Internet of Things, or IoT, is now a reality, even at the minuscule level. With ubiquitous computing, we can stay in constant communication with the outside world. As such there are several types of devices capable of monitoring, measuring, and sending data back and forth from the web. These various devices use different types of communication models; and data models are heterogeneous in nature, and are not interoperable. To have truly seamless communication between heterogeneous devices, an interoperable environment is necessary. We discuss the interoperability problem in IoT devices, and its related issues and identify the research gaps, present a problem statement, and propose objectives and methodology that serve as a guide for future work.

Keywords: Internet of Things, interoperability, data analytics, wireless sensor networks, cybersecurity

INTRODUCTION

The world is seeing a rapid growth in technological paradigm. With ubiquitous computing, embedded computing, wireless sensor networks and advancements in networking technologies, it has become relatively easier to develop, deploy and use devices, that are capable of monitoring, measuring and sending data to and from the web. These devices are termed as 'things' giving rise to the term 'Internet of Things' (IoT) in which everything is connected to the internet. Kevin Ashton first used the term "internet of things" in 1999. Even so, he liked to refer to things as the "internet". The original idea was to have everything equipped with a radio frequency identification system which allowed the thing to be connected to the internet and allow for communication between them and amongst the things. The term "internet of things" refers to the fact that there are more objects linked to an internet connection than there are people. The things to people ratio was 0.08 in 2003 i.e., for every 100 people there were 8 IoT devices. According to Juniper Research's latest report, the number of IoT devices by 2024 will be 83 billion rising from 35 billion in 2020 which accounts for a rise of 130% over 4 years.

IoT has an extensive set of applications which are divided into consumer, industrial, commercial and infrastructure spaces. The application domains range from healthcare, home automation, agriculture, transportation, industry, food, military, maritime, energy management, environment monitoring. The explosive growth of IoT devices in various domains has led to a very diverse and heterogeneous IoT landscape where the devices are heterogeneous with respect to communication technologies, syntax and semantics, hardware and applications.

***Author for Correspondence**
Chaitali A. Chate
E-mail: chate.chaitali@gmail.com

[1-3]Student, Department of Computer Science, KLS Gogte Institute of Technology, Belgaum, Karnataka, India

Received Date: July 27, 2023
Accepted Date: September 01, 2023
Published Date: October 23, 2023

Citation: Chaitali A. Chate, Sanjeev S. Sannakki, Vijay S. Rajpurohit. An Interoperability on the Internet of Things (IoT): A Review. International Journal of Radio Frequency Innovations. 2023; 1(1): 40–53p.

PROBLEM OF INTEROPERABILITY

Connecting two platforms is relatively easy. The challenge is to interconnect several platforms into an existing ecosystem to provide seamless interconnectivity services to the end user. The ecosystem should also enable future technologies to be included for inter-connectivity as well as provide

for legacy systems to connect too. Different vendors may have developed IoT products using different IoT and non-IoT technologies, but cooperation and communication between the different platforms should be made possible irrespective of the underlying technologies. Since most of the vendors are specific and adhere strictly to their company norms, it is unlikely that any of the companies will be willing to adapt their products and processes to suit the needs of the others. Absence of a *de facto* standard in the IoT area becomes a hindrance in realizing globalized IoT standards. The global network infrastructure of IoT devices is growing exponentially and each solution has its own infrastructure, data formats, devices, and APIs. Thus, we are presented with the following issues because of lack of interoperability:

- Device incompatibility: Non-compatible IoT devices cannot be plugged into other non-compatible devices.
- Vendor locks in.
- Difficulties in developing cross platform or cross domain applications.
- Difficulties in seamless sharing of resources.
- Vertical silos.

A comprehensive survey on the state-of-the-art solutions for providing interoperability between different IoT platforms is given by Mahda *et al.* [1]. According to them, the issue of IoT interoperability can be viewed from different perspectives:

- *Device interoperability:* The devices differ in capability, with high end devices, which have higher processing power, storage capacity, energy requirements and communication capabilities or are low end devices which are resource constrained. Since every vendor has their own set of policies, guidelines and use cases; the different vendor devices differ substantially into the various aspects of implementing communication protocol stack. Hence the devices not only differ in terms of resources on board but also in terms of communication technologies that connect them to the Internet. Bringing such devices together to communicate and exchange information seamlessly refers to device interoperability.
- *Network interoperability:* Numerous communications networks are used by the various IoT devices to send and receive information. Some may use the short-range networks while others may use the long-range networks. Hence to provide seamless data exchange between such heterogeneous device's issues related to networking such as addressing, routing, QoS, security, resource optimization and mobility support should be addressed.
- *Syntactic interoperability:* Syntactic interoperability issue arises when the sender's and receiver's basic grammar or syntax is different. That is, an application with XML coding cannot decode JSON documents. They differ in schema, data formats and interfaces used. The coding and decoding rules need to be the same, or a translator is needed which does the translation.
- *Semantic interoperability:* Due to the differences in the syntactical structures, the informational model and data model used by different IoT structures will be different. This difference in the interpretation of the meaning of the syntax refers to semantic interoperability. One device cannot interpret correctly and meaningfully what the other device is communicating.
- *Platform interoperability:* Platform interoperability arises due to the fact that different vendors use different types of operating systems, programming languages, data structures and access mechanisms to develop their applications. Each platform has its own data model, information model and APIs. Vendors, due to various policy issues, prefer using a particular platform. Hence the developers are presented with interoperability problems.

An IoT-specific classification is provided by ETSI and AIOTI which defines four levels of interoperability that includes:

- *Technical interoperability*: deals with communication protocols and their infrastructure required.
- *Syntactic interoperability*: deals with data formats and their encodings.
- *Semantic interoperability*: deals with the common understanding of the data and its interpretation.
- *Organizational interoperability*: deals with the ability of the different organizations to intercommunicate effectively across different systems and infrastructures [2].

IoT Platforms and Interoperability

According to Gartner, an IoT platform is a software suite or platform as a service (PaaS) that allows customers to create apps on the platform that are commonly used to monitor, manage, and control different kinds of end points. It facilitates operations involving IoT endpoints and integration with enterprise resources [3].

Thus, an IoT platform connects the 'things' with the Internet and with the people and applications which consume the data produced by the things. An IoT platform offers functionalities from the hardware and devices to network connectivity, to the data processing and cloud services, to apps and services to the end users and developers. The platforms also provide data ingestion, storage and analytics which helps the developers to develop apps and services using simple APIs provided by the platform. This allows the developers to focus only on the application without worrying about the internal hardware, network, and storage structures. In view of the above functions expected from a platform, an IoT platform should have the following capabilities:

- Provisioning and management of devices and their application software.
- Data aggregation, integration, transformation, storage, and management.
- Event processing.
- Customizing and building applications.
- IoT data analysis and visualizations.
- Cybersecurity.
- Communications device based on IoT.
- Adapter communications.
- User interfaces for end users and developers [4].

IoT platforms are middleware solutions that provide the various above-mentioned functions possibly coming from different vendors, which work together as a single unit in connecting the devices to the people and includes the sensors and actuators, the gateway device, the communication network, data analytics and end user app services.

Identification of the many components of an IoT ecosystem and their interactions is necessary to ensure interoperability. The elements can be broadly classified as the application or service, the underlying platform, application domains, the IoT devices. The interaction patterns can be outlined as:

- Cross-platform pattern: the applications can access information across different platforms using the same interface. Hence the focus here is on having common interfaces or a translation among the interfaces to provide interoperability.
- Cross-application domain access: the applications can access information across different platforms and different domains. Hence here the focus is on providing not only common interfaces but also in providing semantically common data which can be accessed across different domains.
- Platform independence: the application can be hosted on different platforms. The focus here is on developing platform independent applications.
- Platform scale independence: the different platform scales (server-scale, device-scale, fog-scale) are hidden from the applications.
- Higher level service façade: the services act as platforms. Therefore, creating services that are compatible with other services is the main goal here.
- Platform to platform: allow direct interactions between different platforms [5].

Semantic Interoperability

The general meaning of the term semantics is the study of meanings or relationship of meanings [6] it involves the study of the words, signs or symbols involved. When we extend the same semantics to the computer science field, this relationship needs to be interpreted and processed by machines. Hence the knowledge of the words/symbols/signs involved, and their relationships need to be machine readable

and should be in a consistent state such that the meaning does not change from machine to machine. Using ontologies to accomplish this is a simpler method. A formal, explicit definition of concepts in a discourse domain is called an ontology [7]. Together with the concepts, their attributes, restrictions, relations, and instances, they form a knowledge base which can be viewed as an information model in the computer science world.

Tim Burners Lee proposed the idea of the semantic web in 2001 through the term web of data or data web, where the meaning of the data on the web is machine readable. The World Wide Web Consortium (W3C) has created guidelines for what is now known as Web 3.0, an expansion of the World Wide Web. The W3C offers technologies such as Resource Description Framework (RDF) which is a lightweight data model for describing ontologies in the form of meta data and SPARQL, a query language to interact with the meta data.

To achieve semantic interoperability, the systems must have some form of common understanding of the information being shared. This can be achieved by having the systems conform to a common standard of communication by having a single information model or by having a mapping between the different systems' internal information models. Between these two possibilities there exist multiple approaches to achieve semantic interoperability. This paper lists the main approaches as [8]:

- Core information model: all systems conform to a single information model for understanding of the exchanged data. This is easy to implement for system as well as application developers. The hindrances to having such a system are that not all platforms may conform to a single information model. It is difficult to scale the system as the data increases. Those systems whose internal information model does not fit will be excluded.
- Multiple pre mapped core information model: in this approach well established information models (like semantic sensor network ontology or one M2M ontology) are made standards to which the other system's internal core information models are mapped onto. It is a much more flexible and widely used approach since new core information models and mappings can be added over time. But may exclude those systems whose information models do not match the existing ones. Also, this requires the need to provide the mappings for each information model.
- Core information models with extensions: in this approach only a high-level minimalistic core information model is defined which different platforms must conform to. In addition, extensions are provided which can be platform specific. A semantic mapping must be provided when the platforms need to communicate through the extensions. Thus, this provides a flexible approach as the systems conform to a set of core functionalities while having flexibility in defining semantic mappings between the custom extensions. But the defining of the core information model and semantic mappings can be a complex task.
- Pre mapped best practice information model: in this approach, each information model is considered as a core model, hence platforms need not conform to any of the information model. The platforms may choose to be compliant with any of the models. But this requires extensive semantic mapping defining process when platforms are not interoperable.
- Mapping between different platforms: in this approach, every platform is independent and there is no standard. To provide interoperability, semantic mappings need to be defined which can be a complex task.

LITERATURE SURVEY

True interoperability can only occur if everybody gets on the same page, which is a highly improbable and unlikely situation due to reasons such as privatization, market monopoly, ease of use, business goals, financial gains, ever growing technological advancements which provides the developers and companies with multiple options to develop their products and services. In this situation, the interoperability issue is handled by having different approaches that can be categorized based on the underlying approach used in dealing with the problem of interoperability. On a broad level, the approaches to interoperability can be categorized as:

- Providing an intermediate gateway/adapter.
- Virtual/overlay network solutions.
- Networking technologies: IP based, SDN based, NFV, Fog computing.
- Open API.
- Semantic web.
- Service oriented architecture (SOA).
- Open standards [1].

There are several solutions proposed by researchers in industry and academia. We provide a comprehensive literature survey that covers solutions proposed at both the academia and industry level which encompasses different technologies as given above focusing more on the semantic level of interoperability.

A platform called VICINITY offers a range of services and technologies that make it possible to combine different IoT infrastructures into a single, shared framework that allows them to function together [9]. To facilitate the development of IoT applications and the uniform management of IoT objects, or Things, it provides a single language for IoT devices and services. This involves logging in and out of the peer-to-peer network, carrying out automatic discovery, automatically registering, reading, or changing property values, executing actions, managing events, and more. It provides cloud-based interoperability as a service. It uses a common information model, the ontology network, to describe objects semantically. In general, an item can be a tool or a service [10]. MicroPnP VersaSense's Micro Plug and Play (microPnP) platform offers a cohesive set of hardware and software tools that integrates highly reliable wireless networking with an extended battery life, along with automated recognition and configuration of embedded sensor peripherals into the Internet of Things. It provides Plug-and-play peripheral integration with zero configuration. All linked integrated sensing and actuation peripherals are automatically recognised by MicroPnP, which then installs the appropriate device driver software. Low-power passive electrical components are used in this method. It makes it possible to quickly repackage any current IoT actuator or sensor as a MicroPnP-peripheral. All associated device driver software is immediately requested by the network gateway and installed over-the-air when a peripheral is inserted into a MicroPnP device. The peripheral is now completely operational and open for end users to engage with remotely. The network manager provides RESTful APIs to query all devices and immediately push information from sensors into the cloud, further accommodating application developers.

MicroPnp uses open standards and protocols, particularly the IPSO (Internet Protocol for Smart Objects) data model to achieve data interoperability between different devices. Khaled *et al.* proposed the IoT-DDL (Device Description Language) to generate a run time representation of the thing on the thing itself to allow ad-hoc interactions and interconnections [11]. The IoT-DDL attempts to describe the thing's entities, resources, services, attachments, services, device management and communication protocols. They also propose the Atlas thing architecture which is a light weight architecture that defines the tools to identify things on the network. The device description files or configuration files are created manually and uploaded on the things and on the Atlas thing architecture which is present on the server in the cloud. When they are connected, the thing identifies itself through the DDL files, and advertises its services and functions in its smart space. The DDL files serve as a valuable repository to identify the things to identify its services and properties. To self-discover itself, a thing needs to have its DDL files on board, which allows it to advertise its services and capabilities to the space in which it exists. The creation of the DDL files is done manually which has the future potential to be an automated process.

SEMIoTICS deals with the interoperability issue in the IoT domain [12]. The framework utilizes widely used technologies like RDF, RDFS, OWL, WSDL and WoT's Thing descriptions to provide the interoperability descriptions, annotation, and services. They also provide a common and general API model for communication between the different middleware platforms. Every IoT device is given in the

WoT Standard Thing Description (TD) format which is semantically annotated using iot.schema.org. The thing description is then serialized using the JSON-linking data format on which tools for processing and querying data are present. Thus, the common format used to model the data is TD/JSON-LD. This data is stored in the Thing repository. Transformation rules are used to transform data between two different systems based on the tags carried by the data, for which regular expressions or Drool's pattern engine could be used to perform the actual transformation based on the information carried in the tags. The application is demonstrated in a smart sensing environment. Nawaratne *et al.* presented a gateway present in the cloud that receives inputs from IoT devices and then generates actionable insights based on unsupervised learning techniques [13]. Unsupervised learning networks within a cloud network perform predictive analysis of the pre-fed data. Based on past learning experience, the gateway controller generates actionable insights by observing data in real time. Unsupervised learning networks allow for the achievement of data interoperability. Further, these networks can self-generate and learn incrementally. The approach focuses on data interoperability. In this approach, as the computation intelligence is situated in the cloud, it increases response time, which becomes critical in case of emergencies. As the gateway triggers actions in case of emergencies, its reliability and accuracy are very important.

An autonomous enabled gateway is suggested by Rahim *et al.* as an edge computational intelligence [14]. The approach involves having an intelligent controller for data interoperability both at the cloud and at the edge. File conversions between XML and JSON (Python program) are done to provide interoperability. A module named DeepCoder is used for device management. They proposed approaches to providing autonomous device management and data interoperability at the edge of the network. The paper highlights the importance of using approaches to optimize software at the edge gateway. As the interoperability module performs the translation of messages from different application languages like XML and JSON, this would need $n\times(n-1)/2$ conversion codes for 'n' languages which would constrain the computational power of the edge gateway controller. Sebastian *et al.* presented a HTTP/REST service modelled as a Markov Decision Process [15]. A client agent learns the semantics of the HTTP/REST protocol using Q learning method of the reinforcement learning. An agent is modelled as an MDP in HTTP to provide REST services using Q learning method of reinforcement learning. The MDP agent, as claimed by the authors, can be a client or an API gateway. It adaptively learns the syntax and semantics of the HTTP REST protocol and can be used to provide application specific goals. This approach opens up avenues to extend the use of artificial intelligence techniques to other levels of interoperability. It is observed that developing a MDP state transition is not scalable for huge number of resources. Further, the agent requires to know the syntax and semantics of the language involved, it does not learn automatically. The Q learning always starts at the same initial state which may not always be the case. Sebastian *et al.* have developed a natural language processing based deep reinforcement learning algorithm that can interpret the XML and JSON formats by treating them as strings and converting between the two formats [16]. It provides semantic interoperability. Requires no manual pre-processing or feature extraction. Though the work is restricted to conversions between XML and JSON formats, in a limited sense, this work can be extended to other similar formats for providing an interoperability interface. An IoT-SIM paradigm that enables semantic interoperability among heterogeneous IoT devices in the healthcare arena is proposed by Jabbar *et al.* [17]. The data exchange between the physician and the patient is annotated semantically using the RDF format. The RDF graph's data is extracted via a SPARQL query. The system is cloud-based. The SWE (sensor web enablement) framework is used to provide web services to IoT devices. Data received from patient IoT device is identified, classified, and annotated either manually or automatically through RDF and stored in the cloud for the doctors' reference. It provides semantic interoperability. Their work can be extended to perform the annotation of data from sensors into an RDF model that can be programmed to be done automatically. Thierry *et al.* proposed a natural language processing to deliver interoperability in simple things environment [18]. The authors have developed a simple things environment on the Azure cloud platform and provide interaction between humans and machine by using a natural language processing system that works with the concept of web-hooks and APIs. It provides interoperability on the higher level i.e., on a human interaction level, for which English

language was chosen. The translation services can be extended to devices, gateways and IoT platforms and not just restricted to cloud services. Security aspect can also be included in the communication and translation services. Guo *et al*. have suggested an AI based semantic IoT (S-IOT) hybrid service architecture which supports heterogeneous devices and applications based on AI in a smart city application [19]. The proposed platform has three layers: infrastructure layer that supports different types of IoT platforms; resource provision layer that connects the different IoT platforms to different services through semantic modelling based on AI; the semantic layer that obtains the semantic model from the device provider and then constructs and stores the instance of the device according to the semantic model. They proposed two main issues related to semantic mappings: how different devices understand each other and how to effectively mine the associations between semantic models. The proposed platform and unified model can be used as a reference and direction to work on; however, they do not provide suggestions as to how the issues will be tackled by them. Sulyman *et al.* focused on the differences between the various standards used for the IoT development [20]. They elaborated on the architectural challenges in scaling IoT and highlighted mainstream approaches in interconnecting devices across IoT infrastructures both physically and semantically. They proposed an architectural approach to bridging Internet based IoT networks with cellular based IoT systems. For this purpose, they proposed the inclusion of a DOI module in interconnecting the cellular based IoT and Internet based IoT. For connecting to the internet and cellphone IoT nodes, the DOI module manages generic IoT data access. The DOI module performs the following functions which enable any IoT node to communicate with any other IoT node across the cellular and Internet network. The functions of the DOI module are to interface the two systems and enable rapid discovery, identification, authentication, access, and communication. The authors claim that the DOI module, which they propose to introduce at the middleware layer, can be updated to include more standards and hence provide heterogeneous access. In their white paper, Pedro and Skarmeta describe internet of things (IoT) network as an inherent heterogeneous network because of the functionalities and capabilities of each device in it [21]. Thus, they surmise that direct adaptation of the common protocol i.e., the IP network to the internet of things network negates the heterogeneity factor at the network level. The writers support the notion of software defined networking (SDN) because of its adaptability and network programmability. SDN allows the IoT network to be heterogeneous while allowing a higher network controlling entity. They propose a common architecture which consists of a IoT controller on top of an SDN controller. The IoT controller is connected to various IoT devices through the IoT agents present on the devices. Each device is registered with the IoT controller through its IoT agent. When a communication is initiated by a requester to the responder, the IoT agent on the requester device sends this request to the IoT controller. Next, the IoT controller locates the two endpoints in the network, calculates path using some routing algorithm and then communicates the same to the SDN controller. The SDN controller establishes communication paths which can be logical or virtual by setting up forwarding rules in all the forwarding elements. This arrangement has the advantage of having an IoT overlay network which is built on top of SDN networks. It also allows the devices to be heterogeneous.

Bedhief *et al.* combine SDN and docker techniques with centralized SDN controller to face the heterogeneity issues in IoT [22]. Docker is an open platform that allows to create, ship and test applications by allowing applications to be separated from the infrastructure. By using a combination of SDN and docker techniques, the authors claim to achieve both network heterogeneity and device heterogeneity. Their experimental setup consists of client, server and docker hosts connected through ethernet connection with a central POX SDN controller. The network is simulated over a Mininet Emulator. The SDN controller uses OpenFlow through OpenvSwitch which is an open flow virtual switch. They claim to have maintained connectivity between any type of hosts for any type of service. Network heterogeneity is handled by the centralized SDN controller. Established in 2012, oneM2M is an international collaboration initiative by eight of the top ICT standards development organisations worldwide [23]. The organization's mission is to develop international technical standards for M2M and IoT technology interoperability with regard to architecture, API specifications, security, and enrolment solutions. oneM2M standards comprise a horizontal architecture in the form of a 3-layer model

comprising applications, middleware services and networks. It uses open standards, oneM2M REST APIs, gateways to convert between non oneM2M to oneM2M. it supports application protocols like RESTFUL HTTP, CoaP, MQTT. It supports cellular, Zigbee, Bluetooth and Wi-Fi connectivity. It provides security services like authentication, end to end security and privacy management. It provides interoperability at device, network, syntactic, semantic, cross domain, cross platform levels through the use of open standards, open APIs, ontologies and gateways. There are several open source projects that have been using OM2M, OCEAN, OS-IoT, openMTC, IoTDM, OASIS SI, oneM2Mtester.

OMA lwM2M [24]: The LwM2M protocol, a communication protocol from the open mobile alliance provides communication between a device equipped with lwM2M agent and lwM2M servers. It has a contemporary architectural design based on REST, defines an extensible resource and data model, and builds upon the Constrained Application Protocol (CoAP), an effective secure data transfer standard. It is intended for remote management of M2M devices and related service enablement. LwM2M has been specified by a group of industry experts at the OMA SpecWorks Device Management Working Group and is based on protocol and security standards from the IETF. It offers services for device management and service enablement in IOT and M2M devices. It supports XML data format and CoaP application protocol. It supports cellular, Zigbee and Wi-Fi connectivity. It does not provide any security services. It offers interoperability at the device and network level. It carries a small code footprint and hence is most suited for resource constrained devices. The communication between the devices and servers are enabled through a lwM2M client on the device and the servers. The clients are exposed via a standardized data model. The data model is standardized by the IPSO alliance as the object model in which an object is defined for each data concept that is accessible via the lwM2M client.

A lightweight, open JSON-based hypermedia catalogue format for exposing collections of URIs is called HyperCat [25]. Any number of URIs with any number of triple assertions about them that resemble RDF may be exposed by each HyperCat catalogue. HyperCat is easy to use and enables resource descriptions to be published in linked data format for developers. To address a genuine requirement for interoperability throughout eight consortia, HyperCat was developed. It was financed by the Technology Strategy Board of the United Kingdom. The program's main goal was to discuss interoperability, and, how data centres in various sectors may cooperate with one another. The goal is to enable applications to utilise distributed data repositories (data hubs) in tandem by enabling queries to be made of their catalogues in a standard machine-readable manner.

This makes it possible to create “knowledge graphs” of accessible datasets from several hubs, which applications may use to find and query the data they require, regardless of the hub where it is stored. To enhance data discoverability and interoperability, it is a standard for providing Internet of Things data hub catalogues using web technologies. It uses open standards and open APIs to provide syntactic compatibility. It supports JSON and RDF data formats. It uses RESTful HTTP for communication protocol. It does not provide security services. AllJoyn technology was promoted by Qualcomm in 2011 [26]. Devices can interact with other devices nearby thanks to this software framework. It is available as open-source software on the GitHub repository which the developers can download and develop products and services on different platforms like Linux, Windows, iOS, and other real time light weight operating systems. It uses the client-server communication model to function.

It provides an open-source API structure to access the underlying network stacks like Wi-Fi, NFC, Zigbee and Bluetooth. This modularity helps to extend the connectivity to different network stacks. It consists of two types of devices a router node and an application node. Both are implemented using the AllJoyn client library which comes in two flavours: a standard library that is used for higher operating systems and a thin client for deeply embedded systems. Alljoyn provides services like service advertisement and discovery and communication occurs over a D-bus [27]. It supports XML, JSON and EXI data formats and provides interoperability at the device level using open APIs and on-board client software.

The Open Connectivity Foundation (OCF) is funding the open-source software project IoTivity, which enables device-to-device connectivity [28–30]. The foundation is responsible for developing standards, guidelines and providing certifications to support the development of IoT. IoTivity is a free and open-source reference implementation of the OCF protected IP device infrastructure that offers device-to-device and device-to-cloud connectivity through IP at a communication layer that is secure. IoT interoperability is achieved through the use of standard consensus-based data models. The IoTivity stack consists of five horizontal layers comprising the application layer on top of standardized data model layer which is on top of the OCF core framework layer which is built on top of the network and hardware layers. The OCF core framework layer provides services like device commissioning, headless configuration, cloud connectivity and bridging framework. By directly interacting with the non-OCF device over a bridge platform with numerous entity handlers, an OCF device can communicate with a non-OCF device. The entity handlers that are launched by the bridge platform upon startup identify non-OCF devices and generate resources for each one. These resources are then made discoverable by the bridge platform. The entity handler then binds itself to that resource. Clients can then discover and communicate with that resource through the entity handler which performs the mapping between OCF and non-OCF data.

INTER-IoT is financed by the Horizon 2020 initiative of the European Commission, a project that aims to provide interoperability among different IoT platforms. The INTER-IoT project offers an open cross layered solution that provides interoperability at all IoT layers viz., device, network, middleware, application and data level and also provides service like security, device management, virtualization, QoS, service integration, storage, and network communication. It is the first approach which proposes to provide universal translation among platforms. It suggests a device gateway that permits all forms of data forwarding at the device level. The physical layer is decoupled from the virtual layer of the gateway. Thus, the physical part enables forwarding of any type of data while the virtual part handles API and middleware requests. At the network level, it proposes to use SDN and NVF to accommodate any types of networks. At the middleware layer, it proposes to use mediators, brokers and bridges and a general API that allows global exploitation of smart objects across multi-platform IoT systems.

It suggests offering a general API at the programme and services layer to facilitate the creation of new apps. At the semantic layer, the Inter Platform Semantic Mediator (IPSM) is proposed to perform platform-to-platform ontology translations of the information using ontology alignments. For every IoT artefact that wishes to cooperate, communicate, and interoperate, specific OWL-demarcated semantics must be defined.

An open-source programme called Thingspeak enables users to converse with internet-enabled items. It provides access of data, retrieval, and logging of data by providing API to both devices and websites. It is a cloud platform that uses RESTful HTTP and MQTT protocols. Any type of embedded device can be programmed with the API key to send and receive the data to and from the Thingspeak cloud which can be retrieved and analysed using Matlab analytics.

AGILE: Adaptive Gateways for dIverse muLtiple Environments (AGILE) is a project that builds a modular hardware and software gateway for the IoT. It offers hardware-level gateways for a range of wired and wireless networking protocols. At the software level, it gives the community of IoT software developers access to open-source code via the Eclipse Foundation. Software components are isolated from one another and from the underlying hardware using Docker containerization. All the modules in the architecture aim to provide syntactic interoperability at hardware and software levels. It does not focus on semantic interoperability.

OBSERVATIONS AND DISCUSSIONS

After a review of the literature, the observations made are summarized below:

- IoT developers need to write the ontologies and adapter code for their IoT devices and/or services using the platform specific APIs. is, for every platform, they must specify the 'thing description' describing the properties, interfaces, communication protocols etc. using platform specific tools. These are then integrated into the platform. The adapter code is written in the target platform language, e.g., JSON for VICINITY platform which is open source [9]. The adapter code then acts as the interface between the specific IoT devices/services and the specific platform hence providing 'interoperability'. This requires additional work on behalf of the developer, as, if the device is to be interoperable with a different platform, then the developer must write specific adapter code for the new platform. Hence this gives rise to a situation of 'vertical silos' both in industry and academia. These require the developers of IoT to use the specific APIs and data models and definitions for integrating the IoT devices into the required platform.
- The standards available for providing semantic interoperability need the developers to write the ontologies/descriptions for their products. Manual ontology writing is an extensive and time-consuming process which requires domain knowledge.
- Performing the mappings between different systems either at data level or semantic level using AI techniques is an extensive process requiring $n\times(n–1)/2$ number of mappings when there are 'n' number of different systems. Having a common standard to which all systems could be mapped onto would reduce the number of mappings. Ontologies are a step towards this type of mapping.
- Cloud services may increase the response time for time critical applications, hence edge/cloud-edge computation can offer a better option.

PROBLEM STATEMENT

The IoT area is growing globally at an exponential rate with each solution having its own infrastructure, data formats, devices, APIs, communication technologies. The existing platforms and frameworks provide IoT device developers to develop applications and integrate their devices and solutions into their platforms and frameworks by providing their own set of APIs, communication technologies, data and semantic models. This gives rise to vertical silos, vendor lock-in, difficulties in seamless sharing of resources, difficulties in developing cross platform or cross domain applications and hence device incompatibility. Absence of a *de facto* standard in the IoT area becomes a hindrance in realizing globalized IoT standard.

It is expected that interoperability should happen seamlessly and automatically between systems without specific programming. It should not be restricted to mere translations between the information and data models of different systems. As such, a horizontal interoperability between systems should be possible even when new systems are added or new devices come into the network, where there will be no ambiguity regarding the data that is being exchanged and interpreted.

Hence a solution is to be developed that automatically detects, identifies, and integrates an IoT device, constructs a device profile in terms of data model and semantic model and automatically annotates the data generated by the integrated device based on the standard semantic model. This would generate a repository of semantically annotated data which can be accessed by any other device across systems and used across different application domains.

PROPOSED OBJECTIVES AND METHODOLOGY

In order to realize a framework that provides identification and interoperability on the Internet of Things (IoT), the following research objectives are proposed:

- *RO (1):* To discover, identify and configure IoT devices and build a 'Device Description Profile'.
- *RO (2):* To develop an ontology model for inter-domain interoperability based on standard ontologies.
- *RO (3):* To develop an artificial intelligence based automatic annotation system for device data annotation.

- *RO (4):* To implement the automatic annotation system in a relevant inter domain application to justify the veracity of the proposed research.

The following methodology is proposed to achieve the previously stated objectives and provide a solution to the problem statement.

1. To identify the devices in the network and build a device description profile, mobile agent technology may be used.
2. To use a combination of natural language processing and deep learning techniques to automatically annotate the device data in the proposed 'interoperability ontology model' in the standard RDF triple store from unstructured text.

The proposed network topology is as given in Figure 1. The network topology consists of a main central controller like a SDN Controller in the cloud. At the edge there are several sub-controllers which are connected to the IoT gateways at the edge. The IoT gateways connect the individual sensors to the rest of the Internet.

The sensor device identification methodology is depicted in Figure 2. The mobile agent will be dispatched from the sub-controller to the gateway where the sensor device identification module is executed by the mobile agent.

After information retrieval, the mobile agent goes back to the sub-controller where the profiling of the device and its registration in the local database is done based on the information retrieved by the agent. This is depicted in Figure 3.

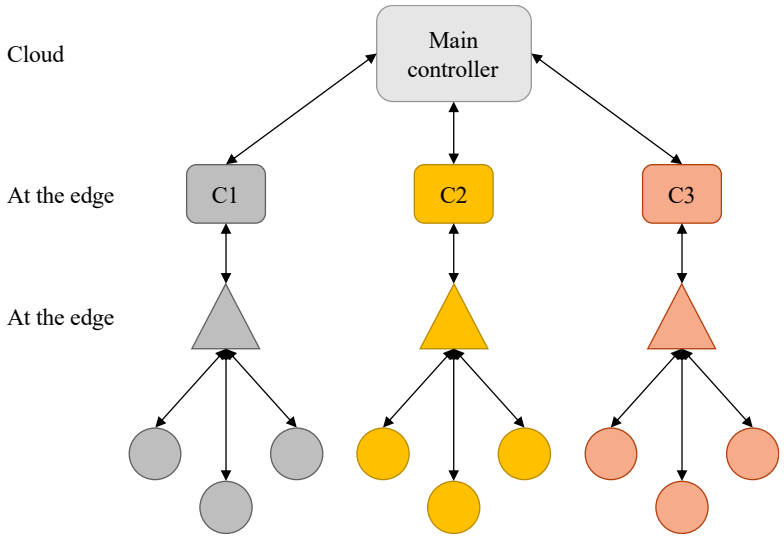

Figure 1. Proposed network topology.

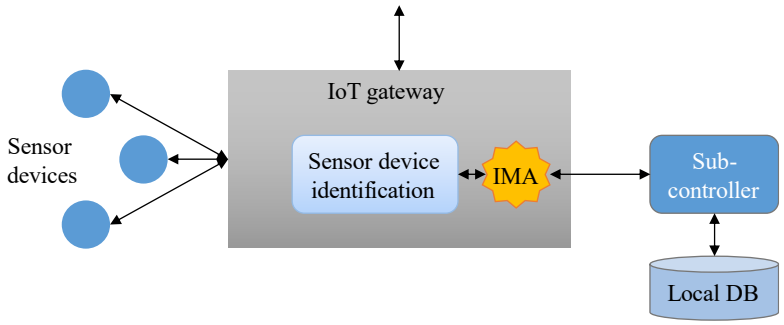

Figure 2. Sensor device identification.

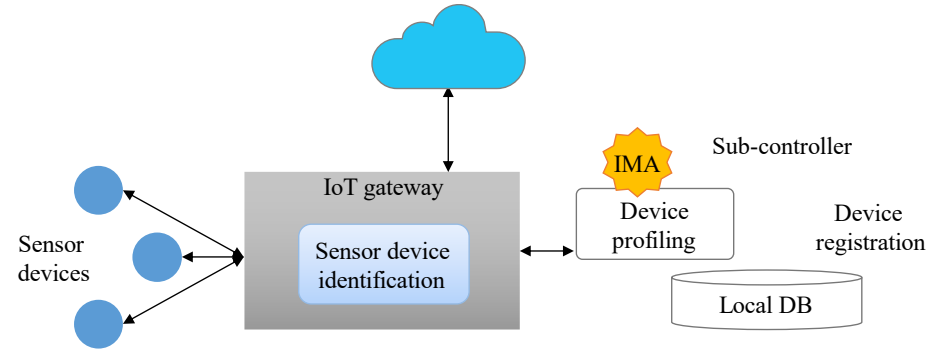

Figure 3. Sensor device profiling and registration.

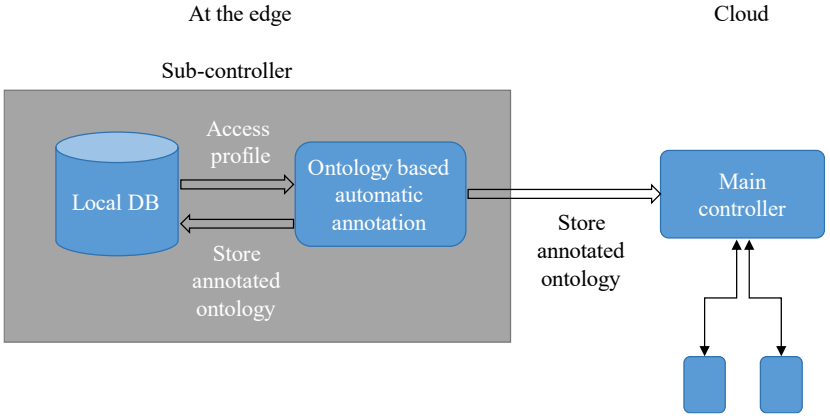

Figure 4. Ontology based automatic annotation.

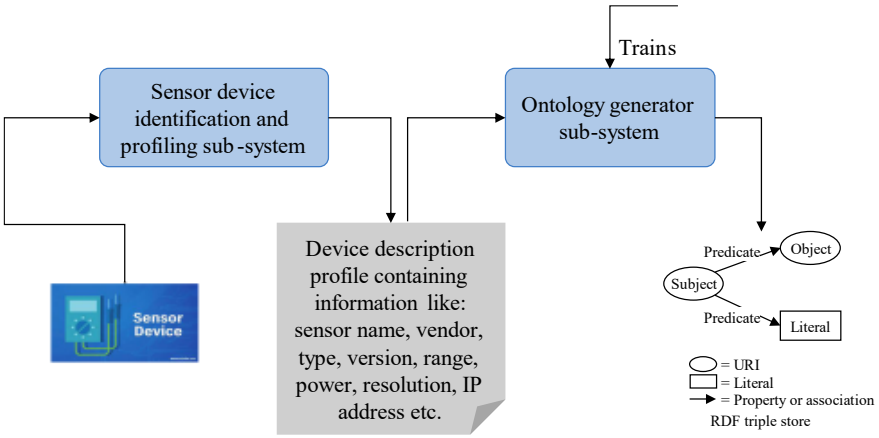

Figure 5. System model.

After the device profiling and registration stage, the data generated by the devices is automatically annotated based on the inter-domain ontology designed. This is depicted in Figure 4.

The system model is given in Figure 5. A sensor device is given as an input to the system for which a device description profile is generated. Next, the device data is automatically annotated based on the pre generated ontology model.

CONCLUSION

In this study, the problem of interoperability in Internet of Things was discussed. The different reasons that contribute to the issue of interoperability was presented. Next, the different types of interoperability that commonly occur in the field of IoT was discussed along with the different category

of approaches that are used while addressing the issue was presented. A comprehensive review of literature was given which included the work done in both industry and academia to address the interoperability issue. The gaps were identified which have the potential to be improved. A problem statement was presented. Based on the problem statement, objectives and detailed methodology was proposed on how the interoperability issue in IoT can be addressed, that may serve as a starting point for working on a solution that provides a more comprehensive view on the issue of interoperability in IoT.

REFERENCES

1. Noura Mahda, Mohammed Atiquzzaman, Martin Gaedke. Interoperability in Internet of Things: Taxonomies and Open Challenges. *Mob Netw Appl.* 2019; 24(3): 796–809.
2. van der Veer H, Wiles A. Achieving Technical Interoperability – the ETSI Approach. *ETSI White Paper No.3. 3rd Edn.* France: European Telecommunications Standards Institute; 2008 Apr.
3. Velosa A, Natis YV, Pezzini M, Lheureux B, Goodness E. *(2015 Jul 2). Gartner's Market Guide for IoT Platforms.* [Online]. Available at: https://www.gartner. com/doc/3086918/market-guide-iot-platforms
4. Vermesan Ovidiu, Jacoby Michael. Advancing IoT Platforms Interoperability. New York: River Publishers; 2018. 10.13052/rp-9788770220057
5. Bröring A, Schmid S, Schindhelm C-K, Khelil A. Enabling IoT Ecosystems through Platform Interoperability. *IEEE Softw.* 2017; *34(1):* 54–61.
6. Merrian-Webster. Merrian-Webster dictionary. [Online]. Available at: https://www.merriam-webster. com/dictionary/semantics.
7. Noy Natasha. Ontology Development 101: A Guide to Creating Your First Ontology. Stanford, CA: Stanford University; 2001.
8. Jacoby Michael, Antonić Aleksandar, Kreiner Karl, Łapacz Roman, Lampert Jasmin. (2017). Semantic Interoperability as Key to IoT Platform Federation. Interoperability and Open-Source Solutions for the Internet of Things. 2016; 3–19. 10.1007/978-3-319-56877-5_1.
9. Zivkovic Carna, Guan Yajuan, Grimm Christoph, editors. IoT Platforms, Use Cases, Privacy, and Business Models: With Hands-on Examples Based on the VICINITY Platform. Springer; 2020 Sep; 227.
10. Matthys N, Yang F, Daniels W, Joosen W, Hughes D. Demonstration of MicroPnP: The Zero-Configuration Wireless Sensing and Actuation Platform. 2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). 2016; 1–2. doi: 10.1109/SAHCN.2016.7732982.
11. Khaled AE, Helal A, Lindquist W, Lee C. IoT- DDL–Device Description Language for the "T" in IoT. *IEEE Access*. 2018; 6: 24048–24063. doi: 10.1109/ACCESS.2018.2825295.
12. SEMIoTICS Deliverable D4.4 Semantic Interoperability Mechanisms for IoT (first draft). (n.d.). Available at: https://www.semiotics-project.eu/wp-content/uploads/2021/01/SEMIoTICS-D4.4_revised.pdf.
13. Rashmika Nawaratne, Damminda Alahakoon, Daswin De Silva, Prem Chhetri, Naveen Chilamkurti. Self-evolving intelligent algorithms for facilitating data interoperability in IoT environments. Future Gener Comput Syst. 2018; 86: 421–432. ISSN 0167-739X,
14. Rahmani Rahim, Firouzi Ramin. Gateway controller with deep sensing: learning to be autonomic in intelligent internet of things. Int J Commun Netw Distrib Syst. 2021; 26(1): 1–29. 10.1504/IJCNDS.2021.111631.
15. Kotstein Sebastian, Decker Christian. Reinforcement Learning for IoT Interoperability. *2019 IEEE International Conference on Software Architecture Companion (ICSA-C)*, Hamburg, Germany. 2019; 11–18. 10.1109/ICSA-C.2019.00010.
16. Klöser Sebastian, Kotstein Sebastian, Reuben Robin, Zerrer Timo, Decker Christian. Deep Reinforcement Learning for IoT Interoperability. In: Advances in Automotive Production Technology: Theory and Application. ARENA2036. Berlin, Heidelberg: Springer Vieweg; 2021. 10.13140/RG.2.2.36301.26087.

17. Sohail Jabbar, Farhan Ullah, Shehzad Khalid, Murad Khan, Kijun Han. Semantic Interoperability in Heterogeneous IoT Infrastructure for Healthcare. Wirel Commun Mob Comput. 2017; 2017: Article ID 9731806(10p). https://doi.org/10.1155/2017/9731 806
18. Thierry G, Zoraida C, David G, Michael M, Debopam B. Natural Language for an Interoperable Internet of Simple Things. 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland. 2019; 474–479. doi: 10.1109/WF-IoT.2019.8767215
19. Guo K, Lu Y, Gao H, Cao R. Artificial Intelligence-Based Semantic Internet of Things in a User-Centric Smart City. Sensors (Basel). 2018; 18(5): 1341. Published 2018 Apr 26. doi:10.3390/s18051341
20. Ahmed Iyanda Sulyman, Oteafy Sharief MA, Hassanein Hossam S. Expanding the Cellular-IoT Umbrella: An Architectural Approach. IEEE Wirel Commun. 2017 Jun; 24(3): 66–71.
21. Martinez-Julia Pedro, Skarmeta A. Empowering the Internet of Things with Software Defined Networking (WHITE PAPER). 2014.
22. Bedhief I, Kassar M, Aguili T. SDN-based architecture challenging the IoT heterogeneity. 2016 3rd Smart Cloud Networks & Systems (SCNS), Dubai, United Arab Emirates. 2016; 1–3. doi: 10.1109/SCNS.2016.7870558.
23. oneM2M Sets Standards For The Internet Of Things & M2M [Internet]. Onem2m.org. 2023 [cited 2023 Oct 17]. Available from: https://www.onem2m.org/
24. OMA LwM2M - Brief description — Anjay 3.6.0 documentation [Internet]. Github.io. 2017 [cited 2023 Oct 17]. Available from: https://avsystem.github.io/Anjay-doc/LwM2M.html
25. https://iot.ieee.org/newsletter/january-2016/hypercat- resource-discovery-on-the-internet-of-things.html
26. OCF - AllJoyn [Internet]. Open Connectivity Foundation (OCF). 2018 [cited 2023 Oct 17]. Available from: https://openconnectivity.org/technology/reference-%20implementation/alljoyn/
27. https://www.slideshare.net/alexgonzalezgarcia/introduction- to-alljoyn
28. OCF - IoTivity [Internet]. Open Connectivity Foundation (OCF). 2020 [cited 2023 Oct 17]. Available from: https://openconnectivity.org/technology/iotivity/
29. IoTivity. (2023). IoTivity. [online] Available at: http://iotivity.org/.
30. Openconnectivity Foundation. (2023 Nov). OCF Bridging Framework Specification. https://openconnectivity.org/specs/OCF_Bridging_Specification.pdf

