Overview on Internet of Things (IoT) Architectures, Enabling Technologies and Challenges

Kaoutar Hafdi*, Abderahman Kriouile, Abdelaziz Kriouile
IMS Team, ADMIR Laboratory, Rabat IT Center, ENSIAS, Mohammed V University, Rabat, Morocco.

* Corresponding author: Tel.: 00212670512710; email: kaoutar.hafdi@gmail.com
Manuscript submitted March 4, 2019; accepted June 4, 2019.
doi: 10.17706/jcp.14.9.557-570

Abstract: IoT systems are known for being complex, heterogeneous, distributed, resources constrained, integrating probably moving devices or services in a highly dynamic environment. This is a non exhaustive set of characteristics that an IoT system should satisfy. Designing an IoT system according to a given architecture in order to satisfy a set of required characteristics is a priority in order to meet principal requirements of the system according to the specific application field. In this paper, we give an overview of main existing categories of IoT architectures. We identify principally software IoT architecture, hardware IoT architecture, and general IoT architecture. Based on this study, we propose an end-to-end IoT architecture designed according to a five layered model. We propose a summary of different enabling technologies presented according to the end-to-end architecture. We finally exhibit main challenges of IoT systems that can be raised at different contexts and applications.

Key words: IoT architectures, software architectures, hardware architectures, distributed systems, IoT systems, IoT challenges, enabling technologies.

1. Introduction

Connecting all everyday objects is a quickly growing fashion nowadays, commonly known as Internet of Things (IoT) systems. IoT systems require a deep understanding of many technological domains. In fact, it is the meeting of various concepts leading to a transversal paradigm. In the literature, there exist a lot of works proposing new IoT architectures applied on a specific or many application domains. Other works focus on technologies used to meet IoT systems requirements from a technical and practical point of view.

We need to deal with architectures in order to represent, organize and structure the internet of things in a way that allows it to work properly [1]. IoT architecture may be treated as a system or paradigm which may consist physical objects (e.g. sensors, actuators), virtual objects (e.g. cloud services, communication layers and protocols) or a hybrid of these two perspectives [2]. These architectures must be able to support IoT devices and services, as well as the workflow that these devices will affect. Thereby, IoT architectures are classified as hardware architectures, software architectures, process architectures and general architectures [1], [3].

In this paper, we propose an overview of existing IoT architectures classified according to three categories (hardware, software, general). According to this classification, we deduce a general end-to-end IoT architecture that can be used to design IoT systems. In addition to that, we list the enabling technologies used in IoT systems. We give main challenges encountered in IoT systems in general.

Outline: The rest of this paper is organized as follows. Section 2 discusses different types of hardware IoT architectures. Section 3 presents software IoT architectures. Section 4 exhibits general IoT architectures. Section 5 shows the proposed End-to-End IoT architecture. Section 6 discusses the enabling technologies used in IoT systems. Section 7 discusses challenges related to IoT systems. Section 8 gives concluding remarks and directions of future work.

2. Hardware IoT Architectures

Many hardware architectures have been proposed to support the distributed computing environment required by the Internet of Things. Among these architectures we find the peer-to-peer architecture, the EPC based architecture and the Wireless Sensor Networks (WSNs) based architecture. In the following, we discuss those three hardware architectures and we give examples for each type.

2.1. Peer to Peer Architecture

In this model, the interaction between processes/peers/devices is symmetric : each process will act as a client and a server at the same time (acting as a ’servant’). Peer-to-peer architectures can be built using a distribution protocol such as the multiple Distributed Hash Table (DHT) routing protocol. It is possible to design a P2P architecture to be especially beneficial for Web of things (WoT) applications, like M2M communication, involving embedded devices [4].

2.2. EPC Based Architecture

EPC(Electronic Product Code) is a universal identifier that gives a unique identity to an item a RFID tag is affixed to. The identity is made to be unique so that each object is identifiable within the objects field [5]. The EPC number enables data information exchange among companies and their business partners. For standardization purposes, an architecture known as the EPCglobal network was proposed. In this architecture, roles, interfaces and a common vocabulary are specified, leaving implementation details for end users depending on the application domain. Such a network allows traceability of the movement of items among the supply chain and gathering information related to each item [6].

Many proposed IoT architectures are based on EPC approach. An EPC based architecture could be built over an heterogeneous access network, particularly using a ZigBee network as it can collect the latest information about ‘Things’ [7]. The proposed architecture provides two functions. The first one is how to register new objects or devices to a home area network. The second one is how to make objects communicate through the Internet with generic protocols. The proposed EPC architecture uses combination of sensor networks and EPC networks, which provide product information through web services from the manufacturers.

Another existing IoT architecture based on EPC approach is the Bridge project [8]. BRIDGE (Building Radio Frequency Identification Solutions for the Global Environment) is an european project that aims to research, develop and implement tools to enable the deployment of RFID and EPC Network applications. This project is implemented in a complete decentralized architecture enabled by EPCglobal network architecture. This architecture is intended for supply chain products and is applied to seven application domains. A main issue while designing such an architecture is security and privacy. Illicit use of EPC and secure transmission of data between readers and tags should be ensured. No extension of the EPC Network standard to deal with sensor data is provided.

2.3. Sensors and WSNs Based Architecture

Wireless sensor networks (WSN) allow embeded devices to be connected and used in a seamless way. Using WSN while desingnig IoT architectures is very promising since it helps implementing distributiveness and context-awareness which are main features of IoT architectures. It is possible to implement an integrated framework for interconnecting WSNs and actuators to standard networks as Web services [9]. While designing an IoT architecture based on WSNs, it is possible to include a complete IP adaptation method, as it is the case for the Sensor Networks for an All-IP World (SNAIL) architecture which includes four significant network protocols: mobility, web enablement, time synchronization, and security [10]. Another existing approach relies on using M2M gateway in the IoT architecture based on WSNs. The main idea is to connect different sensors to the M2M gateway for communication with end users or different provided services. This solution can be applied in smart building applications using WSNs. Heterogeneity and security issues are fulfilled using this approach [11]. There exist another possible approach that permit desingning WSN based IoT architecture. It is called autonomic-oriented architecture and consists in implementing an autonomic communication protocol taking into consideration the constraints of wireless terminal available in the Internet of Things, mainly sensors and RFIDs [12]. WSNs are also widely used in smart cities in order to manage smart traffics and mobility [13].

3. Software IoT Architecture

Software architectures are necessary to ensure access and sharing of services offered by IoT devices. There are several approaches to provide application framework for IoT such as SOA, RESTful and architectures based on fog and cloud computing. These architectures focus on services and flexibility and cover Operating systems, IoT middleware, APIs, Data management, Big data, etc. In the following, we discuss SOA based architecture, RESTful architecture and cloud/fog based architecture.

3.1. SOA Based IoT Architecture

Service Oriented Architecture (SOA) is a software architectural style that is commonly built using web services standards. It is also possible to implement SOA using any other service-based technology, such as Jini [14], CORBA [15] or REST [16]. In SOA based IoT architecture, each device is a service consumer and/or a service provider offering services or sharing resources and interacting with service consumers via compatible service APIs (Application Programming Interfaces). SOA technologies enable publishing, discovery, selection, and composition of services offered by IoT devices [17]. Unlike traditional enterprise services and applications, which are mainly virtual entities, real-world services are provided by embedded systems that are related directly to the physical world [17]. In IoT architectures, we can find both types of services according to the application domain (Fig. 1).

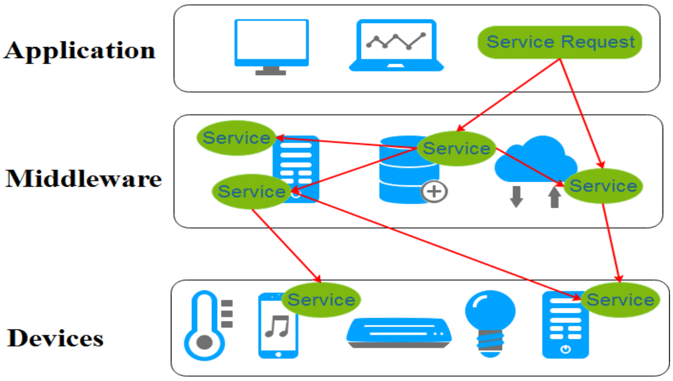


Fig. 1. SOA based IoT architecture.

There has been several works for an effective integration of the Internet of Things in enterprise services using the SOA paradigm [18]-[20]. One existing SOA based IoT architecture is the SOCRADES Integration Architecture(SIA) [19] which is composed of six layers: Application Interface, Service Management, Device Management, Security, Platform Abstraction and Devices. Open and standardized communication via web services at all layers is used. Furthermore, a system architecture is proposed in order to enable dynamic query, select, and use of services running on physical devices. A set of requirements to facilitate the querying and discovery of real-world services is established [17].

IoT systems are heterogeneous and very dynamic. As a consequence, proposing a trust management protocol for SOA based IoT architecture is one open issue [20].

3.2. REST Based IoT Architecture

The Representational State Transfer (REST) is a software architectural style that defines a set of constraints to be used for creating web services. REST architecture is mainly based on constrained client-server communication. REST is implemented by Universal Resource Indicators (URIs) for identifying resources on the Web, Hypertext Transfer Protocol (HTTP), and standardized media types, such as HTML and Extensible Markup Language (XML) to exchange data [16]. The RESTful architecture is known to be loose-coupled, simple and scalable, which motivate to use Web standards to interact with smart things. As a result, the concept of Web of Thigns (WoT) is introduced rather than Internet of Things (IoT) [21]. In the Web of Things concept, smart things and their services are fully integrated in the Web by reusing and adapting technologies and patterns commonly used for traditional Web content. We should notice that HTTP introduces a communication overhead and increases average latency. It should be used for pervasive scenarios where relatively longer delays do not affect the system requirements.

It is possible to build web services for IoT applications by using the Constrained Application Protocol (CoAP) [22], which is defined in the Constrained RESTful Environment (CoRE) charter [23]. the CoAP allows RESTbased communications among applications residing in distributed and networked embedded systems [9], [24]. The CoAP aims to provide a protocol stack able to cope with limited packet sizes, low energy devices and unreliable channels, which are the main characteristics of IoT architectures. As a result, the use of CoAP improves the average of communication latency in IoT systems.

Another existing protocol used to implement the REST architecture is the Message Queue Telemetry Transport (MQTT) [25]. MQTT is a lightweight event- and message-oriented protocol, which allows the devices to asynchronously communicate across constrained networks to reach remote systems. MQTT is based on a publish/subscribe interaction pattern. In particular, MQTT has been implemented for easily connecting the things to the web and support unreliable networks with small bandwidth and high latency. As a result, MQTT is adequat for designing REST based IoT architectures. The main issue with the MQTT protocol is low level security. There exist some work to enhance the security of the MQTT protocol by proposing an Open Source AUthenticated Publish/Subscribe (AUPS) system for the Internet of Things [25].

3.3. Cloud Based IoT Architectures

IoT systems generate a huge amount of data that has to be stored, processed and presented in a seamless, efficient, and easily interpretable way. Cloud computing provide high reliability, scalability, and autonomy to IoT systems. In fact, a cloud based platform acts as a receiver of data from the ubiquitous sensors, as a computer to analyze and interpret data, as well as a visualizations web based tool [26]. It is possible to design an IoT architecture based on a Cloud centric vision [27]. According to this vision, a conceptual framework integrating the ubiquitous sensing devices and the applications is proposed (Fig. 2).

The cloud-centric view is criticized for being very centralized. Indeed, physical devices must be able to communicate with cloud services that are typically geographically remote and scattered. Such an aspect can be very restrictive since the devices used in IoT systems generally have very limited resources. To facilitate access to cloud services, there are architectures that provide the use of access points as intermediates between physical devices and cloud services. This is called the fog computing [28]. The Fig. 3 illustrates an architecture based on the fog computing.

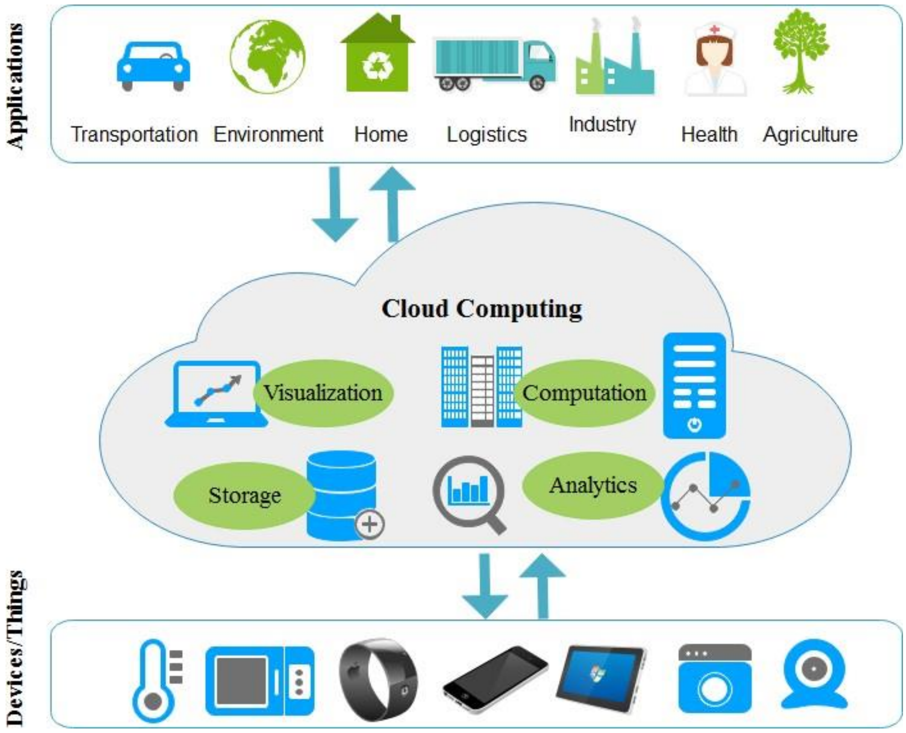


Fig. 2. Cloud based IoT architecture.

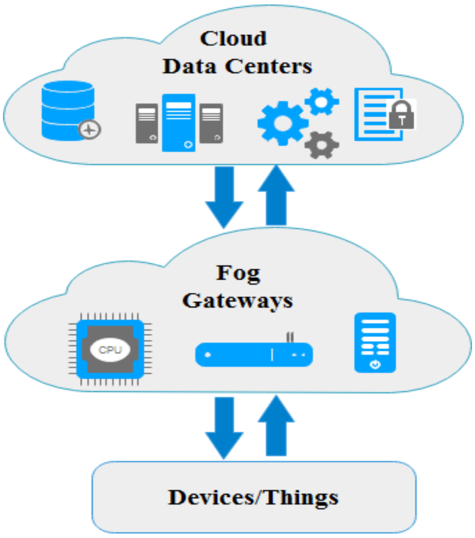


Fig. 3. Cloud/fog based IoT architecture.

4. General IoT Architectures

There is no agreement on a single architecture that best suits the Internet of Things [29], [30]. In the following, we will present general architectures that offer end-to-end solutions covering several aspects of the Internet of Things. These architectures are designed according to a layered model. These layers differ in number and technological solutions deployed depending on the scope and specific requirements for each case.

IoT solutions can be used to design an architecture that is applied on a composite case study consisting of various IoT applications like smart home, smart transportation, smart healthcare, etc. That is the main purpose of researches carried out by [31] which propose a layered and distributed architecture, called

Distributed Internet-like Architecture for Things (DIAT), that provides various levels of abstraction to tackle the issues such as, scalability, heterogeneity, security and interoperability. The functionalities of IoT infrastructure are grouped into three layers (i) Virtual Object Layer (VOL), (ii) Composite Virtual Object Layer (CVOL), and (iii) Service Layer (SL). The three layers are responsible for object virtualization, service composition and execution, and service creation and management respectively. The IoT Daemon module encapsulates the security management transversal module, in addition to the three layers mentioned above.

Applying principals of IoT to smart vehicles results on the new paradigm IoV (Internet of Vehicles). [32] propose a five layered architecture designed for IoV: perception, coordination, Artificial Intelligence (AI), application and business layers. Authors exhibit the protocols that can be used to accomplish the functional requirements of each layer identified in the architecture. In addition to that, a network model of IoV is proposed by identifying major network elements. General and technical recommendations are given to fulfill the IoV requirements using the proposed architecture, especially those concerning the safety, connectivity and location services. A comparison between functional and non functional requirements of IoV and VANETs (Vehicular Adhoc Networks) is provided.

There exist several researches proposing conceptual architectures without being applied on a practical case study. [33] prpose a general IoT architecture with three functional platforms: Sensing and Gateway, Resource and Administration, Open Application. It is an open and generic IoT architecture with open interfaces and resources, considering different business scenarios, application-based requirements, and technologies. This IoT architecture can be applied in nine fields, including: domain industry applications; smart agriculture; smart logistics; intelligent transportation; smart grid; smart environmental protection; smart safety; smart medical care; and smart home. Authors give general recommendations and requirements to deploy this IoT architecture on each field without expliciting a practical case study.

Ref. [34] propose a conceptual architecture for organizing IoT-based pervasive communities and societies. The conceptual architecture is devised to integrate the concept of process-aware collaborative communities into the standardized framework of the Internet of Things by including the concept of process automation. This architecture comprises four layers: IoT application, virtual community support, network, and device layers. No case study is carried out for this conceptual architecture.

IoT solutions can be deployed in logistics and supply chain management services. An IoT-based architecture is proposed for monitoring the conditions in which goods are transported and stored [35]. This architecture includes vehicle fleet tracking, goods monitoring and control and location-based services and aims to improve the performance, safety, and reliability of the transport process. In this proposal, authors give general requirements and directions to implement the IoT-based architecture in transport monitoring systems. A comparative study of technological possible choices is carried out. No practical implementation is handled.

There exist proposals for a generic architecture that aims to be applied to different application fields by giving technological and practical solutions for each case. [36] propose an architecture based on a modular design and subdivided into different layers with several choices for each layer. Choices made on each layer can be replaced or combined without affecting the functionality of the overall system. The different layers in the proposed architecture are: Physical Sensors and Actuators, Low-Power Embedded Processor, Wireless Transceiver Technologies, Internet Gateway and application management cloud server. Taking profit from the modular design, the proposed architecture is applied on several applications such as health care, smart home, agriculture system and object tracking. Technological choices and experimental results are given for each application field.

Summary of IoT architecture:

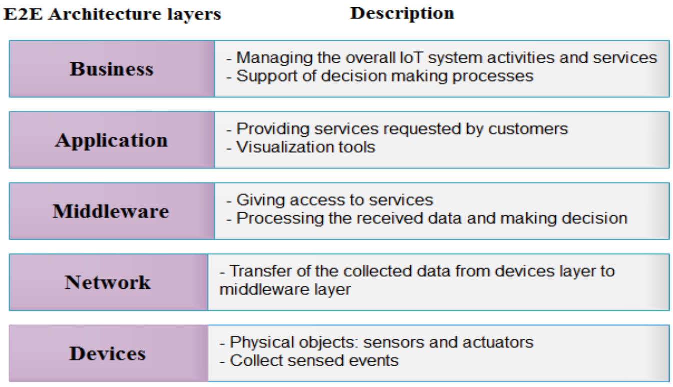
In Table 1 we resume all studied IoT architectures classified according to three principal categories.

Table 1. Summary of IoT Architectures

Hardware IoT Architecture	- Peer to Peer Architecture [4] - EPC based Architecture [6, 7, 8] - Sensors and WSNs Architecture [9,10, 11, 12]
Software IoT Architecture	- SOA based Architecture [17, 18, 19, 20, 37,38] - REST based Architecture [21, 9, 24, 25] - Cloud and fog based Architecture [26, 27, 28]
General IoT Architecture	- Composite case study [31] - IoV application domain [32] - Logistics and supply chain [35] - Conceptual architectures [33, 34] - Generic architecture[36]

5. End-to-End IoT Architecture

From the studied end-to-end architectures, we can deduce a general architecture according to which end-to-end architectures are designed (Fig. 4). This general architecture is composed of five layers: 1) Devices 2) Network 3) Middlware 4) Application 5) Business. In an end-to-end architecture, several layers can be grouped, or a layer can be divided into several sub-layers according to the objectives and requirements of the IoT system studied. In the following we give a description of each layer of this general architecture.



E2E Architecture layers	Description
Business	- Managing the overall IoT system activities and services - Support of decision making processes
Application	- Providing services requested by customers - Visualization tools
Middleware	- Giving access to services - Processing the received data and making decision
Network	- Transfer of the collected data from devices layer to middleware layer
Devices	- Physical objects: sensors and actuators - Collect sensed events

Fig. 4. End-to-end multi-layered IoT architecture.

Devices layer: Also called the perception layer. This layer contains devices in contact with the physical world that can be detectors or actuators. This layer is responsible for grouping the data of the physical world and transmitting them through the network layer. The nature of these devices differs according to the field of application and the nature of the data to be collected: humidity, brightness, movement, heat, etc.

Network layer: This layer is responsible for transmitting the data collected by the perception layer to the middleware layer. Reliable communication channels must be ensured for the transmission of data. There are several technologies that can be used in this layer: Wifi, Zigbee, 3G / 4G, RFID, Bluetooth, etc.

Middleware layer: This layer is responsible for storing and interpreting the collected data in order to route the requested service to the right place. This layer mainly supports the heterogeneity of devices and applications and thus ensures the link between the physical world and the application layer.

Application layer: This layer provides the requested services to the end user through visualization tools. The services provided differ widely according to the application field: smart home, smart factory, smart vehicle, health care, etc.

Business layer: This layer manages the activities and services of the IoT system. It is responsible for studying the data received from the application layer and for structuring and analyzing them in order to organize the results in a dashboard consisting of diagrams, tables, processes, etc., in order to have an overall evaluation of deployed solutions.

6. Enabling Technologies

In this part, we discuss different technologies used in an IoT system. In fact, an IoT system is composed of several elements: the perception of the physical world provided by physical objects in touch with the external environment, the communication of data through adequate means, the storage and processing of collected data and finally the presentation and the visualization of the results by the end users.

To achieve all the elements of an IoT system, there are several technologies that must be combined at different levels in order to implement the features required by the system. Thus, we present in the following technologies to deploy for IoT systems. We rely on the end-to-end architecture previously mentioned (Fig. 4), explaining the technologies that must be deployed at each layer of this general architecture. The Fig. 5 summarizes the technologies deployed at each layer of the general end-to-end architecture. This figure has been realized based on several works concerning the technologies used in the IoT systems [35], [3], [36], [28].

7. IoT Challenges

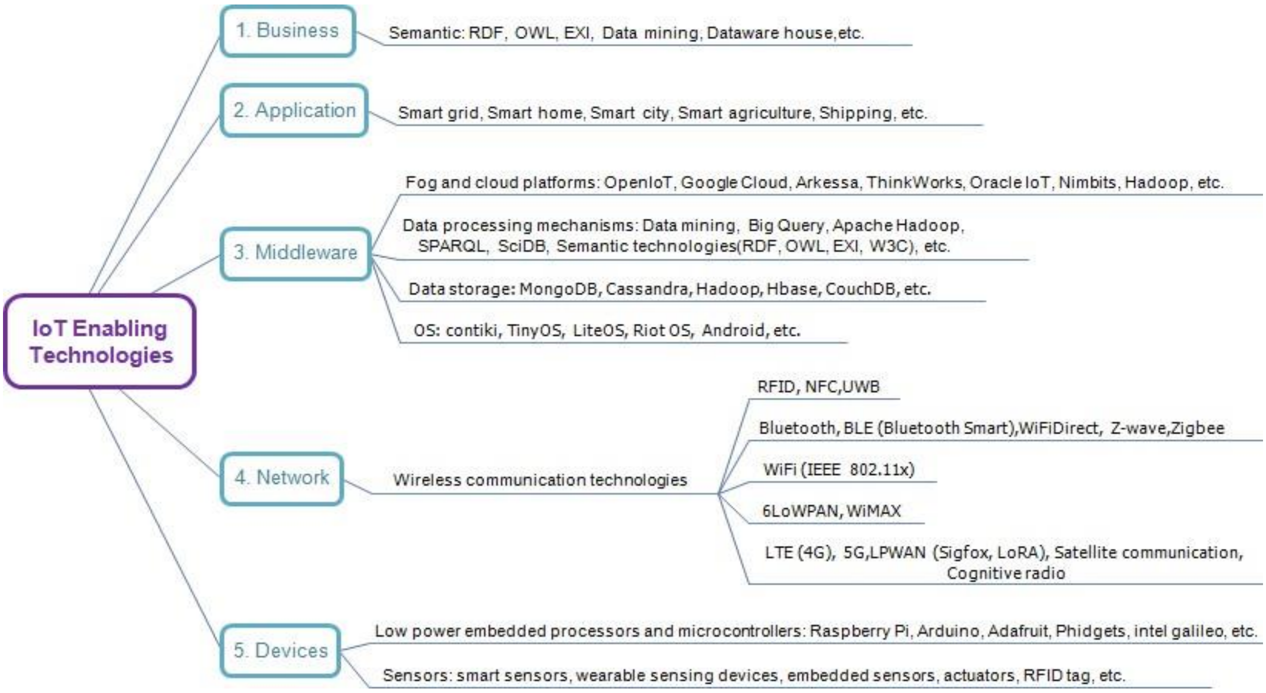

IoT Enabling Technologies
1. Business
Semantic: RDF, OWL, EXI, Data mining, Dataware house, etc.
2. Application
Smart grid, Smart home, Smart city, Smart agriculture, Shipping, etc.
3. Middleware
Fog and cloud platforms: OpenIoT, Google Cloud, Arkessa, ThinkWorks, Oracle IoT, Nimbits, Hadoop, etc.
Data processing mechanisms: Data mining, Big Query, Apache Hadoop, SPARQL, SciDB, Semantic technologies(RDF, OWL, EXI, W3C), etc.
Data storage: MongoDB, Cassandra, Hadoop, Hbase, CouchDB, etc.
OS: contiki, TinyOS, LiteOS, Riot OS, Android, etc.
4. Network
Wireless communication technologies
RFID, NFC,UWB
Bluetooth, BLE (Bluetooth Smart),WiFiDirect, Z-wave,Zigbee
WiFi (IEEE 802.11x)
6LoWPAN, WiMAX
LTE (4G), 5G,LPWAN (Sigfox, LoRA), Satellite communication, Cognitive radio
5. Devices
Low power embedded processors and microcontrollers: Raspberry Pi, Arduino, Adafruit, Phidgets, intel galileo, etc.
Sensors: smart sensors, wearable sensing devices, embedded sensors, actuators, RFID tag, etc.

Fig. 5. IoT systems enabling technologies.

There is still no unanimity for an IoT architecture valid for all areas of application. Each scientific community provides definitions and directions related to its areas of interest. As a result, the challenges of IoT systems also differ depending on the perspectives adopted. In our research work, we will establish various challenges encountered during the implementation of IoT systems (Fig. 6). To do this, we rely on the end-to-end general architecture mentioned earlier. We mention the challenges related to each layer in this architecture. We should notice that there are several cross-cutting challenges that occur on multiple layers but their interpretations differ for each layer (Fig. 7).

Architecture: The realization of an IoT system according to a given architecture is essential. Indeed, this architecture makes it possible to acquire a set of requirements required by the system. Consequently, proposing an architecture for IoT systems is an essential challenge that should be tackled. Indeed, there are several proposals for IoT architectures in the literature. Some of the proposals involve physical architectures, others offer software architectures, while there is another scientific community that offers end-to-end architectures. An architecture must satisfy the functional and non-functional requirements of the concerned system. It is therefore necessary to be able to choose the appropriate architecture according to the objectives and the interests specific to the application field [28], [3], [39].

IoT systems modeling: IoT systems are complex systems because they integrate a large variety of devices and services that should be managed and well integrated. The communication and the interaction between IoT systems components should also be studied. That is why there should be more interest and work on IoT systems modeling issue. Formal modeling of IoT software architecture, hardware architecture, end-to-end architecture as well as the different protocols of communication and security seems to be a promising issue that will enable much control of IoT systems management [3]. In the litterature, there exist works on formal modeling of communication protocols entended for IoT systems [40], [41]. There exists also proposals for modeling IoT systems using Multi Agent Systems [42], [43]. In a previous work, wepropose a formal model for our ReDy architecture entended for IoT systems [44]-[46].

Distributivity: An IoT system is characterized mainly by its distributed aspect. In fact, it inherits the challenges encountered in the design of distributed systems combined with the specific features of IoT systems. In the following, we cite four main challenges related to IoT systems as distributed systems [47], [48], [26], [49].

a) Membership management: This involves making decisions for the choice of method to design the architecture according to which the network is built. This choice must comply with the requirements of the Internet of Things [47].

b) Communication within the distributed network: Choice of the communication protocol in order to meet the requirements of the system [47], [48].

c) Distributed Intelligence: The entities in the system must be able to perform calculations in order to make decisions without external intervention [47], [50].

d) The management of massive data exchanged in the distributed network [47], [26].

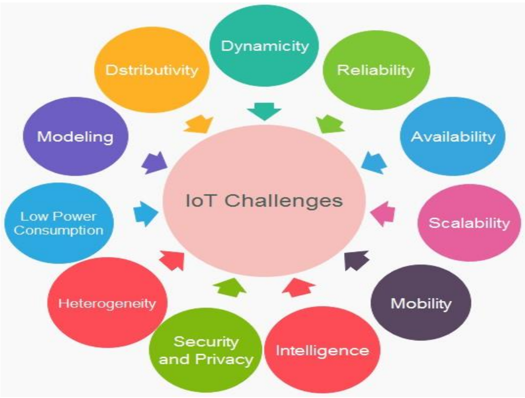


Fig. 6. IoT challenges.

Dynamicity: An IoT system is usually composed of a very large number of entities. During system operation, entities can join or leave the system. Such a change shall in no way affect the functioning of the system that must continue to provide the required services. It is important to ensure the dynamic and potentially migratory aspect of IoT systems [2], [31], [48].

Availability: IoT systems availability must be ensured in both hardware and software levels. By hardware availability we mean that devices in contact with the environment must always be ready to capture the necessary data or event. Software availability means that the services provided by the system must always be accessible by the end user [49], [3], [28].

Reliability: Consists in ensuring the proper functioning of the system while respecting its specifications. Reliability must be ensured at the communication network level by using reliable communication protocols to ensure fault tolerance. Reliability must also be ensured by the hardware components (devices) as well as the software components (services). Reliability and availability are closely related because reliability is about ensuring system availability over time [49], [3], [28].

Scalability: Is one of the most important challenges in IoT systems. Indeed, it is necessary to be able to design modular architectures allowing the extension of the system without affecting its quality of functioning [3], [28], [49], [50].

Heterogeneity: IoT systems are characterized by the heterogeneity of their physical objects, hence the need for a middleware to integrate these heterogeneous devices in a single system. The issue of heterogeneity also concerns the software layers, since the services offered by the system can also be heterogeneous and require standardization work in order to be able to support several technological choices [3], [2], [48]-[50].

Intelligence and decision-making: IoT devices must be connected to the network, and must implement solutions and algorithms in order to be able to make adequate decisions [3], [2].

Mobility: IoT devices are generally small devices that can be moved continuously. Studies are made in order to maintain a correct fonctionning of the system despite the mobility of devices [28], [51], [13]

Security and privacy: It is necessary to have a policy of security and privacy in an IoT system that cover all layers of IoT architecture. In fact, there exist protocols that should be implemented in order to ensure this aspect. However, there still need to adapt existing protocols to the very fast moving context of IoT systems [3], [28], [49].

Low power consumption for computing and communication purposes: IoT systems devices are generally ressources constrained namely in terms of energy capacity. This issue should be taken into consideration while designing IoT systems [47].

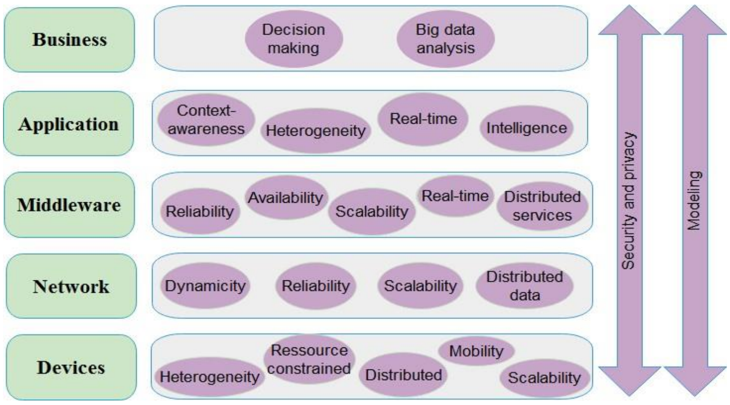


Fig. 7. IoT challenges organized according to the end-to-end architecture layers.

8. Conclusion

IoT systems are the meeting of many technological fields and applications. Tackling such a paradigm needs to have a global view of several aspects related to IoT systems. In this paper, we propose an overview of different existing IoT architectures that allow implementing IoT systems at different layers. From the studied architectures, we propose a general end-to-end IoT architecture composed of five principal layers. We give a summary about the main enabling technologies used to implement IoT systems classified according to the proposed end-to-end IoT architecture layers. We finally exhibit the principal challenges encountered while designing and implementing IoT systems. By this work, we aim to give a global view about IoT systems from an academic point of view in order to be initiated to IoT systems design and implementation. Our proposal can be used in many application fields of IoT systems such as smart home, health monitoring, smart transportation, smart agriculture, etc.

References

[1] Whitmore, A., Agarwal, A., & Da, X. L. (2015). The internet of things a survey of topics and trends. *Information Systems Frontiers*, *17(2),* 261-274.

[2] Ray, P. P. (2018). A survey on internet of things architectures. *Journal of King Saud University-Computer and Information Sciences*, *30(3),* 291-319.

[3] Colakovic, A., & Hadzialic, M. (2018). Internet of things (iot): A review of enabling technologies, challenges, and open research issues. *Computer Networks.*

[4] Andreini, F., Crisciani, F., Cicconetti, C., & Mambrini, R. (2010). Context-aware location in the internet of things. *Proceedings of GLOBECOM Workshops (GC Wkshps)* (pp. 300-304).

[5] Smith-Ditizio, A. A., & Smith, A. D. (2019). Using rfid and barcode technologies to improve operations efficiency within the supply chain. *Advanced Methodologies and Technologies in Business Operations and Management*, 1277-1288.

[6] Gogliano, O., & Cugnasca, C. E. (2013). An overview of the epcglobal network. *IEEE Latin America Transactions*, *11(4),* 1053-1059.

[7] Hada, H., & Mitsugi, J. (2011). Epc based internet of things architecture. *Proceedings of 2011 IEEE International Conference on RFID-Technologies and Applications (RFID- TA)* (pp. 527-532).

[8] Aigner, M. (2010). Bridge building radio frequency identification for the global environment. *Report on First Part of the Security wp: Tag Security*.

[9] Castellani, A. P., Bui, N., Casari, P., Rossi, M., Shelby, Z., & Zorzi, M. (2010). Architecture and protocols for the internet of things: A case study. *Proceedings of 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)* (pp. 678-683).

[10] Hong, S., Kim, D., Ha, M., Bae, S., Park, S. J., Jung, W., & Kim, J. E. (2010). Snail: An ip-based wireless sensor network approach to the internet of things. *IEEE Wireless Communications*, *17(6).*

[11] Fantacci, R., Pecorella, T., Viti, R., & Carlini, C. (2014). A network architecture solution for efficient IoT wsn backhauling: Challenges and opportunities. *IEEE Wireless Communications*, *21(4),* 113-119.

[12] Pujolle, G. (2006). An autonomic-oriented architecture for the internet of things. *Proceedings of IEEE John Vincent Atanasoff 2006 International Symposium on Modern Computing, JVA'06* (pp. 163-168).

[13] Somov, A., Dupont, C., & Giaffreda, R. (2013). Supporting smart-city mobility with cognitive internet of things. *2013 Future Network & Mobile Summit,* 1-10.

[14] Arnold, K., Scheifler, R., Waldo, J., O'Sullivan, B., & Wollrath, A. (1999). *Jini Specification*. Addison-Wesley Longman Publishing Co., Inc.

[15] Siegel, J., & Frantz, D. (2000). *CORBA 3 Fundamentals and Programming*, *Vol. 2.* New York: John Wiley & Sons.

[16] Fielding, R. T., & Taylor, R. N. (2000). *Architectural Styles and the Design of Network-Based Software Architectures,* 7. Irvine doctoral dissertation, University of California.

[17] 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 Transactions on Services Computing*, *(3),* 223-235.

[18] Grønbæk, I. (2008). Architecture for the internet of things (IoT): Api and interconnect. *Proceedings of Second International Conference on Sensor Technologies and Applications, 2008. SENSORCOMM'08* (pp. 802-807).

[19] Spiess, P., Karnouskos, S., Guinard, D., Savio, D., Baecker, O., De Souza, L. M. S., & Trifa, V. (2009). Soa-based integration of the internet of things in enterprise services. *Proceedings of IEEE International Conference on Web Services, ICWS 2009* (pp. 968-975).

[20] Chen, R., Guo, J., & Bao, F. (2016). Trust management for soa-based iot and its application to service composition. *IEEE Transactions on Services Computing*, *9(3),* 482-495.

[21] Guinard, D., Trifa, V., Mattern, F., & Wilde, E. (2011). From the internet of things to the web of things: Resource-oriented architecture and best practices. *Architecting the Internet of Things*, 97-129.

[22] Shelby, Z., Hartke, K., & Bormann, C. (2014). The constrained application protocol (coap). *Technical Report.*

[23] Shelby, Z. (2012). Constrained restful environments (core) link format. *Technical Report.*

[24] Castellani, A. P., Gheda, M., Bui, N., Rossi, M., & Zorzi, M. (2011). Web services for the internet of things through coap and exi. *Proceedings of 2011 IEEE International Conference on Communications Workshops (ICC)* (pp. 1-6).

[25] Rizzardi, A., Sicari, S., Miorandi, D., & Coen-Porisini, A. (2016). Aups: An open source authenticated publish /subscribe system for the internet of things. *Information Systems*, *62,* 29-41.

[26] Abdmeziem, M. R., Tandjaoui, D., & Romdhani, I. (2016). Architecting the internet of things: State of the art. *Robots and Sensor Clouds*, 55-75.

[27] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29(7),* 1645-1660.

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

[29] Fra¨mling, K., & Nyman, J. (2008). Information architecture for intelligent products in the internet of things. *Beyond Business Logistics Proceedings of NOFOMA* (pp. 224-229).

[30] Kortuem, G., Kawsar, F., Sundramoorthy, V., & Fitton, D. (2010). Smart objects as building blocks for the internet of things. *IEEE Internet Computing, 14(1),* 44–51.

[31] Sarkar, C., SN, A. U. N., Prasad, R. V., Rahim, A., Neisse, R., & Baldini, G. (2015). Diat: A scalable distributed architecture for IoT. *IEEE Internet of Things Journal*, *2(3),* 230–239.

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

[33] Chen, S., Xu, H., Liu, D., Hu, B., & Wang, H. (2014). A vision of IoT: Applications, challenges, and opportunities with china perspective. *IEEE Internet of Things Journal*, *1(4),* 349–359, 2014.

[34] Kim, M., Ahn, H., & Kim, K. P. (2016). Process-aware internet of things: A conceptual extension of the internet of things framework and architecture. *TIIS, 10(8),* 4008–4022.

[35] Causevic, S., & Haskovic, A. *The Model of Transport Monitoring Application Based on Internet of Things.*

[36] Yelamarthi, K., Aman, M. S., & Abdelgawad, A. (2017). An application-driven modular IoT architecture. *Wireless Communications and Mobile Computing*.

[37] Giner, P., Cetina, C., Fons, J., & Pelechano, V. (2010). Developing mobile workflow support in the

internet of things. *IEEE Pervasive Computing*, *9(2),* 18-26.

[38] Kawsar, F., Kortuem, G., & Altakrouri, B. (2010). Supporting interaction with the internet of things across objects, time and space. *Internet of Things (IOT),* 1-8.

[39] Stankovic, J. A. (2014). Research directions for the internet of things. *IEEE Internet of Things Journal*, *1(1),* 3-9.

[40] Aziz, B. (2016). A formal model and analysis of an IoT protocol. *Ad Hoc Networks*, *36,* 49-57.

[41] He, F., Baresi, L., Ghezzi, C., & Spoletini, P. (2007). Formal analysis of publish-subscribe systems by probabilistic timed automata. *Proceedings of International Conference on Formal Techniques for Networked and Distributed Systems* (pp. 247–262).

[42] Alexakos, C., & Kalogeras, A. P. (2015). Internet of things integration to a multi agent system based manufacturing environment. *Proceedings of 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)* (pp. 1-8).

[43] Wang, S., Wan, J., Zhang, D., Li, D., & Zhang, C. (2016). Towards smart factory for industry 4.0: A self-organized multi-agent system with big data based feedback and coordination. *Computer Networks*, *101,* 158–168.

[44] Hafdi, K., & Kriouile, A. (2015). Designing ReDy distributed systems. *Proceedings of 2015 IEEE International Conference on Autonomic Computing (ICAC)* (pp. 331-336).

[45] Hafdi, K., Kriouile, A., & Kriouile, A. (2017). Formal modeling and validation of ReDy architecture intended for IoT applications. *International Journal of Innovative Research in Computer Science and Technology*, *5,* 339–349.

[46] Hafdi, K., Kriouile, A., & Kriouile, A. (2018). IoT ReDy architecture for smart grid management. *Computer and Information Science*, *11(4)*, 36-44.

[47] 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.

[48] Taivalsaari, A., & Mikkonen, T. (2017). A roadmap to the programmable world: Software challenges in the IoT era. *IEEE Software*, *(1),* 72-80.

[49] Razzaque, M. A., Milojevic-Jevric, M., Palade, A., & Clarke, S. *Middleware for Internet of Things: A Survey.*

[50] Bojanova, I., & Voas, J. (2017). Trusting the internet of things. *IT Professional*, *(5),* 16-19.

[51] Valera, A. J. J., Zamora, M. A., & Skarmeta, A. F. (2010). An architecture based on internet of things to support mobility and security in medical environments. *Proceedings of 2010 7th IEEE Consumer Communications and Networking Conference* (pp. 1-5).

Kaoutar Hafdi is a Ph.D student at National Higher School for Computer Science and Systems Analysis (ENSIAS), Mohammed V University and IMS team at ADMIR Lab in Rabat IT Center. She received a master of research in parallel, distributed, and embedded systems from Grenoble INP Institute in Grenoble, France obtained in 2013. She received also a master of engineering in software engineering from ENSIAS, Mohammed V University, Rabat, Morocco obtained in 2012.

Abderahman Kriouile is an assistant professor at National Higher School for Computer Science and Systems Analysis (ENSIAS), Mohammed V University, Rabat, Morocco. He is the co-founder of the start up Farasha Systems, which deals with the optimization of the output of solar energy power plants. Dr. Kriouile achieved a PhD in 2015 in Formal Methods applied to the verification of embedded systems on micro-electronic chips. His Ph.D was carried out in the framework of a collaboration between STMicroelectronics and the french national research organization Inria. Prior to that, Dr. Kriouile gained engineering experience in the Avionics and Simulation Department of Airbus in Toulouse, France. Dr. Kriouile holds a MSc. in embedded system and software engineering from Telecom Nancy, France obtained in 2011.

Abdelaziz Kriouile is a full professor in the Software Engineering Department at National Higher School for Computer Science and Systems Analysis (ENSIAS), Mohammed V University, Rabat, Morocco. He is a member of IMS Team at ADMIR Lab in Rabat IT Center. He was the head of SIME Lab. (Mobile and Embedded Information Systems Laboratory) from 2010 to 2017. He received his Ph.D in computer science from the Nancy University, France in 1990. He received a state doctorate from the University of Mohammed V, Rabat, Morocco in 1995. His research activities focus on information systems, cloud computing, and context-aware service-oriented computing. He leads numerous projects related to the application of these domains.