Data Management Mechanisms for IoT: Architecture, Challenges and Solutions

Bassirou Diène[1], Ousmane Diallo[1], Joel J. P. C. Rodrigues[2,3], EL Hadji M. Ndoye[1], Ciprian Teodorov[4]
[1] Department of Informatics, University of Assane Seck, B. P. 523 Ziguinchor, Senegal
[2] Federal University of Piauí (UFPI), Teresina - PI, Brazil
[3] Instituto de Telecomunicações, Portugal
[4] Lab-STICC UMR CNRS 6285 ENSTA, Bretagne, France
bassirou.diene@ucad.edu.sn, {elm.ndoye, odiallo}@univ-zig.sn, joeljr@ieee.org, ciprian.teodorov@ensta-bretagne.fr

***Abstract*— The current traditional database management mechanisms and analytics architectures are not generally suitable for addressing the intrinsic characteristics of diversity, heterogeneity, large-scale, dynamic and large amount of data generated by the Internet of Things (IoT) networks and various IoT applications needs. Then, it is challenging to provide efficient IoT data storage and query processing mechanisms for meeting the requirements of applications. This paper identifies the most relevant characteristics and mechanisms of IoT data management and categorizes them. Moreover, a new four layers fog-based architecture using the distributed approach is proposed for providing a secure storage infrastructure and efficient real-time IoT data discovering with better latency. Finally, this work shows advances on IoT data management mechanisms, highlights their advantages and limits, and discusses the challenging open research issues that need to be focused for providing guidelines for new contributions.**

Keywords—Data management, Internet of Things, IoT data types, architecture, Key challenges, Solutions.

I. INTRODUCTION

The latest progress in Internet with its underlying technologies, smart sensors and new communication technologies, provide the possibility to connect machines, devices, software, and objects communicating among them without human intervention. So, a new paradigm considered as one of the main evolutions of the fourth generation of Internet appeared and is called Internet of Things (IoT).

The IoT network infrastructure is generally considered very large with billions of heterogeneous identifiable IoT devices widely distributed on a given zone and communicating among themselves in order to interact with environment by exchanging sensed data, while reacting to events and triggering actions to control the physical world. The IoT device nodes are considered smart, with autonomous and self-configurable behaviors. However, the IoT devices have generally very limited resources, especially those of storage, processing and energy. This gives rise to, among other problems, the reliability and data validity, heterogeneous data management, performance, security, the privacy[1].

Several research works and efforts are made on IoT technologies, such as RFID technology, sensors and actuators, wireless mobile communication technologies, embedded systems and cloud computing technologies. These advances allow the IoT technologies to bridge the gap between ubiquitous network-based devices and technologies that monitor and collect information from physical world observations and provide new services and diverse applications for improving people's live conditions. Some examples of these applications are smart homes and offices, logistics and distribution systems, healthcare, surveillance and security, supply chain, manufacturing industry, etc. [2] .

Data management is one of the most important aspects in IoT systems. Thus, once the IoT devices perform their data collection, the problem of data secure storing, data discoveringand real-time querying arises. In fact, the diversity of billions of IoT objects, such as sensors, RFID readers, etc. continually provide a large volume of data that have, among other properties, multi-source and heterogeneous, large scale, dynamic, spatiotemporal becoming less accurate over the time, interoperable, contextual and multi-dimensional. The IoT objects are expected to reach 212 billion entities globally deployed by the end of 2020 [2]. In this context, current traditional database management mechanisms and analytics architectures are not generally well suitable. Then, it is challenging to provide a large scale, adequate and efficient IoT data storage and query processing mechanisms for meeting the requirements of various and sophisticated IoT applications. In this paper [3], a study from 68 IoT organizations revealed that: The average IoT organization's total volume of data grew by 30% over the past year, 54% of IoT organizations reported that their current data analysis capabilities are insufficient, 50% of IoT organizations failed to improve time-to-decision over the past year.

Several data and query management approaches for IoT have been proposed, such as middleware-based IoT approaches, data storage approaches, indexing approaches and IoT data schema support approaches [4]. The main contributions of this paper are the following:

- A deep study that presents different characteristics of generated IoT data as well as the main specifications and mechanisms of data and query management for IoT.
- A presentation and classification of the most recent and relevant proposals of data management mechanisms for IoT.

- A proposal of a new four layers fog-based architecture using the distributed approach for a secure storage infrastructure and efficient real-time IoT data discovering with better latency.
- An insight discussion and presentation of the challenging open research issues that need to be addressed for providing guidelines for further contributions.

The remainder of this paper is organized as follows: Section II explores the IoT data types and presents the essential conceptual features of data management mechanisms for IoT. Section III surveys the most recent and relevant data management mechanisms for IoT, while Section IV presents the proposed architecture for IoT data Management. Section V presents an insight discussion on the different proposed mechanisms for IoT data management, and identifies the challenging open research issues that need to be focused for eventual new contributions. Finally, Section VI concludes the paper and pinpoints further research works.

II. Background

Every day, new IoT devices, sensors, and machines come online and feed information from physical world observations into data management systems. Thus, in IoT system, data is one of the most valuable aspects and their study and management play a very important role and become a key research topic. Several types of data can be identified due to the variety and diversity of data sources.

II.1. Features of IoT Data Types

The IoT data have distinctive characteristics due to the data producers and consumers, their diversity, e.g. temperature, humidity, camera, body sensors, RFID readers, etc. and number (about billions of connected objects), the way they are produced, such as in discrete time moment or continuous, automatically generated, etc. and their fields of study, e.g. healthcare, military, transport, supply chain, etc. Thus, These IoT data can be classified as follows [5] [6] (See Fig. 1 for an illustration of features of IoT data types):

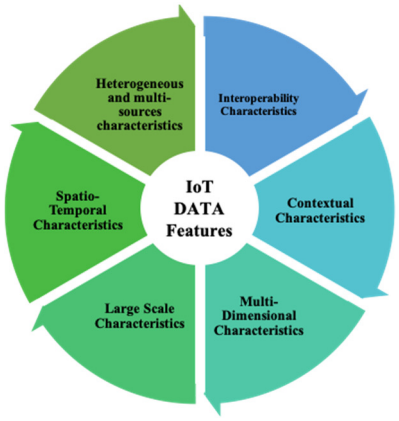

Fig. 1 – Various features of IoT data types.

II.1.1. Heterogeneous and multi-sources characteristics

IoT data is provided by billions of distributed objects of various types, given then this sub-classification:

Radio Frequency Identification (RFID) Data: RFID refers to identification and tracking using radio waves, and it used in many areas, e.g. passports, livestock tracking, road tolling, supply chain management, logistics, stock control, and healthcare. RFID tags can be miniaturized to few millimeters in length and width and inserted into everyday objects and used to transmit and receive data. It is based in an integrated circuit that can store data and an antenna to receive and transmit signals. A tag is activated by radio waves emitted wirelessly from an RFID reader. Once activated, the tag sends data stored in its memory relating to the item back to the reader. RFID tags can be active or passive.

Sensor Data: WSNs are one of the ways used to produce data from IoT systems. Sensors can be distributed in large scale in all sort of geographical area to collect information for users interested in monitoring and controlling a given phenomenon, e.g., camera, weather, temperature, noise, etc.

Positioning Data: It provides location of a particular tagged object either within a Global Positioning System (GPS) or within a local positioning system. GPS includes multiple satellites sending signals to a controlling unit from which objects can establish their position through triangulation. Local positioning systems (e.g., cellular base stations, Wi-Fi access points, and TV towers.) work in a similar way, with smaller coverage. Positional data are very useful in IoT systems, given that IoT objects may be static or mobile.

Metadata about Objects, Processes and Systems: Generally, metadata provides description of data, and is essential to enable users to find and access the appropriate data particularly when managing a huge amount of data, as in IoT systems. IoT objects should be self-describing and have the capacity to report on dynamic characteristics to maximize sharing. It would be important to store data about objects, processes, and systems, so users know how to take advantage of the services and facilities offered by IoT systems.

II.1.2. Large scale characteristics

The IoT objects are expected to reach 212 billion entities globally deployed by the end of 2020 [2]. These devices continually generate data, leading then over the time to a quick expansion of data scale. IoT data storage mechanisms should leverage on new management technologies of huge amount of data, such as cloud, data-center, fog computing to meet the need for IoT data storage and exploitation [7] [8].

II.1.3. Spatio-temporal characteristics

In IoT systems, the data generated by fixed or mobile sensory devices must closely reflect the current state of the targeted environment or phenomenon. However, the environment changes constantly and the data are generally collected in discreet times, thereby leading to the collected data to have spatiotemporal characteristics.

II.1.4. Multi-dimensional characteristics

Generally, WSNs allow receiving IoT data for applications for various purposes, for example healthcare, home building monitoring, military, etc. These applications simultaneously monitor many indicators such as weather, temperature, noise, humidity, light, pressure, etc. Therefore, this leads to the collected sample data to become multidimensional and different aspects must be taken into account in the sampling. For example on the acquisition frequency, such as continuously, at regular intervals or only during a request. Then, once the data is

captured, a problem on how to use or interrogate it can arise, especially when the analysis must be archived in real time. Therefore, it is deemed necessary to propose very efficient techniques for managing this multidimensional characteristic of IoT data [9] [10].

II.1.5. Interoperability characteristics

Data-sharing is very important for IoT systems in order to facilitate collaborative work between different IoT applications. For example, in Internet of Medical Things (IoMT), in addition to physiological data, the traffic information is also important to evaluate the arrival time of the ambulance to decide what kind of first-aid plan is needed.

II.1.6. Contextual characteristics

In IoT environment, the context includes, among other elements, the *Location* which represents the location of the Iot device / clients where the IoT data is generated or consumed. The *Network Condition* which describes the current network condition such as topology and resource (e.g., computation, storage, etc.) available at each node and communication quality, e.g., delay and packet loss between nodes. The *Type of Service* which specifies service's features such as sensing, actuating, computing, and storing which define the possibility of deploying a particular service on a specific node/device based on its resource availability, e.g., a storing service cannot be conducted at a light-weight sensor without storage capacity. The *Quality of a Service (QoS)* which describes the expected QoS when users/consumers receive a service upon their request. For example, the latency can be used as an indicator for QoS.

II.2. Data Management Mechanisms for IoT

Although the relational database management techniques have been deeply studied they are not suitable for managing IoT data due to their distinct characteristics. In fact, in IoT systems there is a large and growing number of heterogeneous data sources like sensors, RFIDs, embedded systems, etc.; While in traditional database systems, the volume of data is collected from finite and predefined sources and stored in scalar form according to strict rules of relationship normalization. Traditional data management systems handle storage, retrieval, and updating of basic data items, records, and files. However in IoT systems, data management systems must efficiently handle online data in real-time while providing storage, logging and auditing capabilities for offline analysis. In IoT system, data is sent continuously from multitude objects to data warehouses, and queries are more frequent with more requirements, while traditional database management systems (DBMS) are subject to occasional queries and updates. Last but not least, in IoT systems obtaining a strict relational database schema and a practice of relational normalization can be relaxed in favor of unstructured and more flexible forms of structures that adapt to different types of data and to different complex queries. However, some aspects of traditional DBMSs, such as remote storage at the object layer, support for unstructured data, relaxation of ACID properties, can be used and adapted for IoT systems.

III. REVIEW ON IOT DATA MANAGEMENT MECHANISMS

Data management mechanisms for IoT have been subject to many proposed approaches in the literature that can be categorized in to four classes: *data storage approaches*, *data indexing approaches*, *sources and data middleware-oriented approaches*, *data/query processing strategies.*

III.1. Data storage approaches

There are two main approaches of IoT data storage which are the distributed and centralized approach [6].

A. Centralized approach

In the centralized approach, the IoT devices act as data collectors. The collected data are sent to and stored in a data center where user/application queries are processed.

The authors in [6] proposed a centralized data storage management system for massive and heterogeneous IoT data called IOTMDB, which is based on NoSQL. The data provided by IoT applications is collected and sent to IOTMDB system. This later includes four main nodes: master node, standby node, data reception node and slave node, where data is received by a data reception node and is stored in a slave node. In their approach, data of various objects are represented in the form of a collection of SampleRecords which is composed of a set of SampleElement defined as follows:

$$<sampleElement>=<key, value>$$

where the key is a String and equals to the name of the value, and the value comes from the union of String and Number and represents the actual sample value.

The authors in [11] proposed a data storage framework that allows efficient storage of both structured and unstructured IoT data. The structured data is managed by a database management module that takes into account relational databases and NoSQL databases [6] , while the unstructured data is managed by a file repository module that extends the Hadoop Distributed File System (HDFS). The main approaches used in the proposed model are object-entity mapping and query adapting method. Object-entity mapping classifies real world objects to entities in databases enabling developers to operate data in databases as they operate the objects in real world. The query adapting module is deployed with many adaptors enabling the processing of some operations such as the joining operations not supported by NoSQL databases.

B. Distributed approach

In this approach, IoT devices are considered as local databases and data are managed locally. This approach exploits the capacities of storage and calculation of devices in order to limit the amount and size of data transmitted.

The authors in [12] proposed a distributed in-networking storage mechanism for discovering IoT data via a machine learning algorithm. A Query Processing (QP) receives user queries and forwards them to Discovery Services (DSs), which has the responsibility for routing and locating a set of related Gateways (GWs) that might have references of connected resources that provides a response to queries.

The work of [13] proposed to dispense to a conventional vision of captured data in a form of stream filtered and continuously aggregated towards a centered storage structure. Sensors are equipped with flash memories and embedded systems to store the data locally on nodes in order to reduce the communication costs in query processing. A DBMS, called StoneDB, takes data from locally stored sensors to create a

database that can support archived queries and even data mining tasks. The transmission is counted only to send requests to the database and when to receive results.

III.2. Data Indexing Approaches

For efficient research and discovery of data and services in IoT networks, continuous scanning of all connected devices appears to be inefficient and computationally intensive. Many data indexing approaches have been proposed [10].

The authors of [14] proposed a distributed indexing approach, which is a discovery services based on Distributed Hash Table (DHT) for RIFD network. Their approach tried to improve the DS in terms of compatibility with various standards RFID and better scalability.

The authors of [12] proposed a novel distributed and efficient indexing mechanism in a hierarchical distributed network to allow discovery of IoT data. In their proposed architecture, indexing IoT resources is performed on a set of distributed gateways. Each gateway represents a cluster of IoT resources that belong to this cluster and has a direct access to them. This approach called semi-distributed is extended to a fully distributed indexing scheme, where there is a tree structure per cluster which has a finite number of children representing the number of types per cluster. Both schemes are able to answer user queries.

III.3. Sources and Data Middleware-oriented Approaches

Designing an IoT system can be problematic because of the difficulty to define and enforce a common standard among all the various devices of diverse domain in. Moreover, in this kind of system, several applications demand abstraction or adaptation layer. For that, a middleware approach can be useful for acting as a bond joining the heterogeneous components together and providing common interfaces and protocol compatibility that allow network-enabled devices to share and publish their data and services seamlessly on a global network [15] .

Most of the current data management proposals are mainly designed for WSNs, which are a subset of an IoT system, and therefore do not explicitly address the distinctive architectural characteristics of IoT. Thus, the authors of [4] proposed a framework that is compatible with the IoT data lifecycle and addresses the design primitives of an IoT data management system. The framework is a middleware-oriented approach centered on data and their sources. It uses the concepts of federated database management systems to ensure the independence of IoT subsystems. The system has a multi-layered architecture that can respond to real-time and archival query, analysis and service requirements.

The work of [5] proposed an architecture which is based on a middleware for data processing in large-scale WSNs in order to hide the network heterogeneity and facilitate data aggregation. It uses a virtual sensor that integrates multiple data streams from real sensors in lower layers into a single data stream. The queries received as input stream are evaluated and the results are stored in temporary relations to be consumed by the application or stored permanently if necessary.

III.4. Data / Query Processing Strategies

Two of the main purposes of collecting data from IoT objects are reporting and doing analysis. However, both involve the pre-data processing at some point in the system for retrieving useful information. Efficient data/query processing strategies based at data location, the intrinsic characteristics of IoT objects, the huge amount of generated data and the storage architecture is a design concern for satisfying various IoT applications. Three processing approaches can be deployed for IoT systems:

The *in-network processing technique* which includes the different types of operations (e.g., aggregation) that are traditionally done on the server side inside the sensor nodes. It is used to filter and reduce the huge and needless data.

The *centralized processing*, on the other hand, requires that data either in its raw form or in an aggregated, more compact form be periodically sent to central persistent storage to enable sophisticated analysis tasks and query processing.

A *hybrid approach* of both techniques can be used for a more flexible processing mode, with various degrees of customization to accommodate the various IoT application needs.

For accessing data in traditional relational systems, querying languages have been used with Structured Query Language (SQL) being their standard. Later, additional constructs have been added to the language for accommodating new forms of data, such as TinySQL for sensor networks and StreamSQL for stream processing. In the context of IoT, a query can be issued either to request real-time data or to retrieve a certain view of the data stored within the system. There are mainly three types of query according to the demands of IoT applications [5]: 1) historical query, to get the data of one time node or a time sequence; 2) tracking query, for example, to get the delivery path of goods of a supply chain; 3) preference query, for example, to get the location of nearest hospital when a man is in danger. As SQL is too complex due to the continuous extensions for new capabilities, it has been suggested a more flexible form be used, in which an SQL dialect or predefined configuration is chosen according to the specific requirements of the scenario at hand.

NoSQL is increasingly gaining popularity because of its functionalities and advantages. The main function of NoSQL is storing and managing unstructured data in the form of key-value. One of the biggest differences between NoSQL and relational DBMS is that NoSQL systems separate data storage and data management, while the RDBMS tries to satisfy both. NoSQL systems support big data storage and are schema-free, with no strong consistency requirements as in relational database systems. For that many query processing mechanism for IoT, based on NoSQL have been proposed [6].

IV. IOT DATA MANAGEMENT ARCHITECTURE

Generally, IoT systems have real-time and latency-sensitive service requirements. Whereas, it is well known that the systems based only on cloud datacenters environment involve high latency in service delivery[7] [8]. So, to provide better latency and Quality of Service (QoS), one can opt for a fog infrastructure based architecture which brings the computing and storage resources to the edge of the network i.e., closer to the IoT devices [16] [17] [18]. The proposed fog-based architecture shown in Figure 2 seeks to improve the IoT data management for a secure storage infrastructure and efficient real-time IoT data discovering with better latency. It is composed of four layers described below, named *IoT Devices Layer*, *Fog Nodes Layer*, *Cloud, Datacenters Layer* and *Application Layer*.

Two kinds of query are considered: the historical data user queries intended to the cloud, datacenters layer and the real-time user queries intended to the IoT devices layer for real-time data.

IoT Devices Layer: This layer is composed of IoT devices or sensors (data resources or producers) which are able to collect data from the real world, process and transmit information to IoT applications through linked gateways. These IoT data producers include among other attributes a type (e.g. temperature), location, value, timestamp and an absolute validity interval (avi) (for real-time processing purpose). The resources are clustered in the gateways according to their data type and their location. To do this, on can use a multi-attribute clustering algorithm [12]

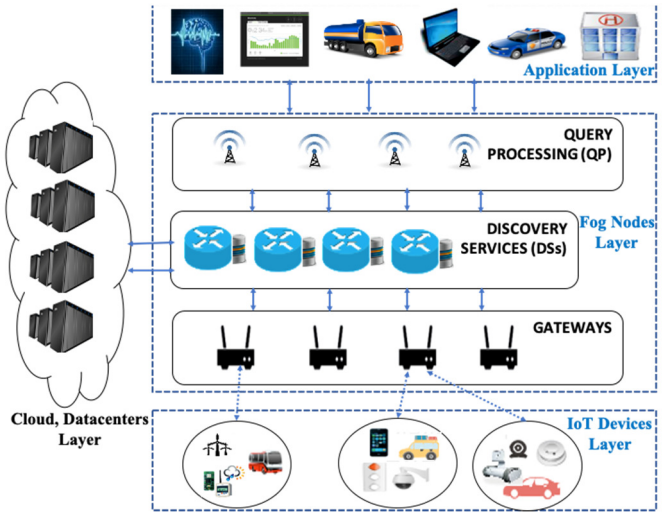

Fig. 2 - Illustration of an IoT Data Management architecture.

Fog Nodes Layer: Generally, a Fog computing infrastructure is composed of traditional networking components, such as routers, switches, proxy servers, Base Stations (BS), etc. and can be placed at the closer proximity of IoT devices. These components have various computing, storage, networking, etc. capabilities and can support service-applications execution. In the proposed architecture the Fog nodes layer includes, among these components, a query processor (QP), discovery services (DSs) and gateways.

A. ***Query Processor (QP):*** the QP processes either historical data user queries or real-time user queries coming from the application layer with respect to query's parameters, such as data type, location and time attributes, and forwards them to the DSs.

B. **Discovery services (DSs):** DSs have the task of controlling and routing historical data user queries to cloud, datacenters layer and the real-time user queries towards the gateways that might know the IoT devices that should have the requested data based to its routing table. In fact, the DSs have a routing table that contains the list of identifications of gateways (GatewayID) with the list of locations of IoT devices belonging to its cluster and their data types.

C. ***Gateways:*** The set of gateways are the cluster heads of IoT devices clusters and have direct access to them. Each gateway has a routing table that contains the list of identifications of IoT devices (IoTDeviceID) belonging to its cluster, their data types and their locations. Thus, when a gateway receives a real-time user query, it checks its routing table and routes the query to the IoT device (s) holding the required data. After computing, the concerned IoT devices reply directly to the initiator of the query.

Cloud, Datacenters Layer: the cloud is a centralized approach with virtually unlimited capabilities in terms of storage and processing power. It is generally used to satisfy IoT requirements, such as ubiquity and high availability. The cloud, datacenter layer is used to store and process the large volume of IoT historical data.

Application Layer: IoT applications, such as smart homes and offices, intelligent transportation systems, smart healthcare service industry (hospitals), smart businesses and industries, safer mining production, firefighting, etc.

V. Discussion and Key Challenges

IoT has known a rapid growth through the number of various applications developed for improving people's live condition. Data management is one of the most important aspects in IoT systems. Thus, it is challenging to provide efficient IoT data storage and query processing mechanisms for meeting the requirements of applications. Indeed, traditional database management mechanisms and analytics architectures are not generally suitable due to the intrinsic characteristics of IoT data earlier detailed in section II.1. While several efforts have been made to address IoT data management issues, there are still many challenges to overcome.

Based on the above detailed analysis of data management mechanisms proposals for IoT, the following open issues can be identified and suggested:

C1: Data storage approach and real-time data processing: Data storage is an important research issue in IoT systems regarding the massive and heterogeneous data generated. Most proposed solutions relies on centralized data repositories, e.g. cloud-based approach, that are storage and queries-intensive or use pre-defined resource links, which make them not scalable for large-scale networks of devices. This approach is considered to be sensitive to create a bottleneck and wastes resources with huge amount of transmitted data, while non all of them are relevant or needed. Moreover, this approach is unsuitable for real-time query processing because it involves time delay for the results. An alternative may be a mechanism based on a distributed approach, which allows data management at the edge of the network with in-network processing for which only end results are needed to be sent. However, this approach processes the data within the network devices which have often limited resources in terms of storage and power and are sensitive to sudden failures. To deal with this, one can opt for a hybrid approach where, in addition to in-network data processing, some supplied data with results may still be kept periodically in a data repository, but are processed in real-time to provide prompt reporting for delay-sensitive queries. Thus, the design of a cloud and fog based framework that takes into account real-time data processing and service provisioning techniques under massive structured or unstructured IoT data, dynamic resources management may well meet the requirements to a generalized database management system to handle various IoT applications and user needs. In addition, based on the above analysis, there is approximately no proposal based on real-time database techniques, which should be useful for IoT applications and may be a promising research topic.

C2: Query representation: IoT data storage components are expected to deal with heterogeneous data resources with unstructured and structured data. Most of the proposals studied in this work interrogates data through SQL-like and/or NoSQL

queries language. Since NoSQL does not support certain types of queries, such as complex joins, it is challenging to propose mechanisms that include this type of query in a distributed and heterogeneous environment integrating structured and unstructured data. Moreover, Extensible Markup Language (XML) offers a means of representing less structured as well as structured data, together with some level of self-description. It is a well-accepted technology that supports interoperability. This technology can be improved for IoT data schema and XPATH / XQuery may be used for flexible and easy queries.

C3: IoT data schema: In database area, the structure of a database system is formally defined with a database schema. In the relational model, schema is defined beforehand as tables and relationships linking those tables, and all data insertions/updates must adhere to that schema. Recently, for a more flexible structure in database management mechanisms that support large and diverse data volumes, researchers adopted non-schema approaches. There are trade-offs between a schema-based approach and having a non-schema solution. The lack of schema leads to records being parsed at run time, as opposed to load-time parsing that is typical in relational database systems. This causes degradation in terms of performance for non-schema systems, as compression becomes less effective. Moreover, with a no-schema approach users will have to write their own parsers, which compromises the interoperability that is desirable for IoT applications. It is challenging to provide efficient query optimizer in non-schema systems, because of the lack of knowledge about indices, table partitions and cardinalities, and statistics about data values.

C4: Metadata management: Efficient metadata management is very useful when managing a huge and heterogeneous amount of distributed data, as in IoT systems. No solution based on metadata was found. The design of a framework based on suitable schemes or non-schema that takes into account the various and distinctive characteristics of IoT data with a metadata management system incorporated in a query optimizer may well meet the requirements of a generalized data storage and querying mechanism to handle various IoT applications and may be promising research topic.

C5: Database security and privacy: IoT allows people and things to be connected anytime, anyplace, with anything and anyone, using any path/network and any service and provides services in different application domains. So, it is question about data from the daily life of human beings. This rises up challenges in data security and privacy. Generally, the choice of appropriate cryptographic methods depends on the processing capability of interconnected smart objects. The traditional security services are not directly applied on IoT due to different communication stacks and various standards. Therefore, the design of flexible security mechanisms adaptable to various IoT applications are needed. Another interesting open research topic is the integration of Blockchain data management [19]. Thus, given the functioning of distributed blockchain and its consensus mechanisms to reconcile divergent interests and distributed trust through the removal of the single trusted third party, blockchain data management for IoT can offer answers to certain security challenges.

VI. Conclusion and Future Works

This work provided an insight presentation and analysis of data management mechanisms for IoT based on relevant proposals in the literature. Then, a new four layers fog-based architecture that should provide a secure storage and efficient real-time IoT data discovering with better latency is provided. Finally, relevant key challenges for well managing IoT data have been identified, discussed for future contributions. The work intended in the near future is the improvement of the proposed architecture with appropriate clustering algorithm and its implementation and analysis.

Acknowledgments

This work has been partially supported by FCT/MCTES through national funds and when applicable c-funded EU funds under the Project UIDB/EEA/50008/2020; by Brazilian National Council for Research and Development (CNPq) via Grant No. 309335/2017-5; by UASZ; by ENSTA.

References

[1] B. Diene, J. P. C. Rodrigues, et al., "Data Management Techniques for Internet of Things", MSSP 138 (2020) 106564, Elsevier, pp.1-18

[2] A. Al-Fuqaha, et al., "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," IEEE COMST. 17 (2015) 2347–2376.

[3] Peter Krensky, "Data Management for the Internet of Things," report research, aberdeen group, Feb. 2015

[4] M. Abu-Elkheir, et al., "Data Management for the Internet of Things: Design Primitives and Solution," Sensors. 13 (2013) 15582–15612.

[5] J. Cooper, A. James, "Challenges for database management in the internet of things," IETE Tech. Rev. 26 (2009) 320–329.

[6] T. Li, Y. Liu, Y. Tian, S. Shen, W. Mao, "A Storage Solution for Massive IoT Data Based on NoSQL," in: IEEE, 2012: pp. 50–57.

[7] R. Mahmud, et al., "Fog Computing: A Taxonomy, Survey and Future Directions," Internet of Everything, Springer (2018), pp. 103-130

[8] X. Wang, et al., "Offloading in internet of vehicles: a fog-enabled real-time traffic management system," IEEE Trans. Ind. Inf. 14 (2018) 4568–4578

[9] H. Zhang, C. Meng, "A multi-dimensional ontology-based IoT resource model," IEEE 5th ICSESS, Beijing, China, 27-29 June 2014, pp. 124-127.

[10] Y. Ma, et al., "An efficient index for massive IOT data in cloud environment," ACM Int. Conf. Inf. Knowl. Manag. (2012), pp. 2129–2133.

[11] Li. Jiang, et al., "An IoT-Oriented Data Storage Framework in Cloud Computing Platform," IEEE Trans. Ind. Inform. 10 (2014) 1443 1451.

[12] Y. Fathy, et al., "A distributed in-network indexing mechanism for the Internet of Things," in: IEEE 3rd WF-IoT, USA, 2016: pp. 585–590.

[13] Y. Diao, et al., "Rethinking Data Management for Storage-centric Sensor Networks," in: CIDR, 2007: pp. 22–31.

[14] P. Liu, et al., "A dht-based discovery service for rfid network," IEEE Int. Conf. Green Comput. Commun. Cyber Phys. Soc., (2014) 344–347.

[15] S. Bandyopadhyay et al., "Role Of Middleware For Internet Of Things: A Study," Int. J. Comput. Sci. Eng. Surv. 2 (2011) 94–105.

[16] A. V. Dastjerdi et al., "Fog Computing: principles, architectures, and applications," Book Chap. in IoT: Principles and Paradigms, Morgan Kaufmann, USA, 2016, p. 15.

[17] S. Sarkar, et al., "Assessment of the Suitability of Fog Computing in the Context of Internet of Things," IEEE TCC., 6(1), p. 46‑59, janv. 2018.

[18] F. Bonomi, et al., "Fog computing and its role in the internet of things," Proc. of 1st workshop on Mobile cloud computing, 2012, p. 13–16.

[19] H. T. Vo, et al., "Research Directions in Blockchain Data Management and Analytics," short paper, in the 21st Int. Conf. on EDBT, 2018