Towards Data Governance in the Internet of Things

Sabrina Sicari, Alessandra Rizzardi, Cinzia Cappiello, Daniele Miorandi, Alberto Coen-Porisini

Abstract The diffusion of Internet of Things (IoT) technologies enables the provision of advanced and valuable services, but also raises several challenges. First of all, the increasing number of heterogeneous interconnected devices creates scalability and interoperability issues and thus a flexible middleware platform is needed to manage all the sources together with all the tasks related to data collection and integration. In fact, the large amount of data has to be properly managed. In particular, on the one hand, data have to be protected from security threats; on the other hand, it is necessary to consider that data are useful only if their quality is suitable for the processes in which they have to be used. For these reasons, it is important that applications/users that aim to exploit the collected data are aware of data quality and security levels in order to understand if data can be trusted and thus used. In this chapter, we present a distributed architecture for managing IoT data extraction and processing that also includes algorithms for the assessment of data quality and security levels of considered sources. A prototype of such an architecture has been realised: through a user interface it is possible to access data services able to filter data from IoT devices on the basis of security and data quality requirements.

Sabrina Sicari
Dipartimento di Scienze Teoriche e Applicate, Università degli Studi dell'Insubria, via Mazzini 5, 21100 Varese (Italy) e-mail: sabrina.sicari@uninsubria.it

Alessandra Rizzardi
Dipartimento di Scienze Teoriche e Applicate, Università degli Studi dell'Insubria, via Mazzini 5, 21100 Varese (Italy) e-mail: a.rizzardi@uninsubria.it

Cinzia Cappiello
Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano (Italy) e-mail: cinzia.cappiello@polimi.it

Daniele Miorandi
U-Hopper, via A. da Trento 8/2, 38122 Trento (Italy) e-mail: daniele.miorandi@u-hopper.com

Alberto Coen-Porisini
Dipartimento di Scienze Teoriche e Applicate, Università degli Studi dell'Insubria, via Mazzini 5, 21100 Varese (Italy) e-mail: alberto.coenporisini@uninsubria.it

The chapter describes the prototype and shows some experiments performed by using several real-time open data feeds characterized by different levels of reliability, quality and security.

1 Introduction

The diffusion of Internet of Things (IoT) technologies turns everyday objects into *Smart Objects*. A global network infrastructure allows such physical objects or *things* to interact among themselves and with the environment where they are placed, in order to fulfill a given goal [1]. Such an interaction, together with the collection and integration of sensed data supports, brought to the provisioning of innovative and customized services to individuals and businesses in different application domains, but also raises many challenges. In fact, the resulting system may include an extremely large number of heterogeneous devices and thus creates scalability and interoperability issues. Hence, it is necessary to deal with the variety of protocols, domains and applications and with the fact that estimations state in 2020 the number of Internet-connected things will reach 20 billions. Moreover, connecting physical objects to the Internet implies the transfer and management of a high amount of data that need to be properly governed. In particular, in this chapter, we aim to highlight that data governance concerns the analysis of data processes and risk management and, in particular, it includes the assurance of regulatory compliance, security, privacy, and quality.

Security & privacy are widely acknowledged to represent critical issues in such a context [2]. Besides security issues related to the communications over the Internet, it is also necessary to consider that devices have limited interfaces to monitor network intrusion and they have data vulnerabilities that can often be exploited to attack the system. In this scenario, the confidentiality and the integrity of the transmitted and stored information has to be guaranteed, and authentication and authorization mechanisms have to be provided to prevent unauthorized users or devices to improperly access the system. Furthermore, taking into account that data are often related to personal and/or sensitive information, privacy of users, intended as the ability to support data protection and users anonymity, has to be ensured [3].

As stated above, the large amount of available data enables the design of smart services. However, in order to obtain valuable results, it is necessary to consider that not all the values might be relevant: errors, missing or outdated values can negatively affect decisions [4] [5]. Data quality represents another essential requirement for the adoption at scale of IoT services: provided results should be correct and reliable or at least the users should be aware of both security and quality level of the data being accessed, in order to take informed decisions about their usage.

For these reasons we claim that an effective data governance should be supported by a system able to manage heterogeneous data sources and to evaluate the security and the quality of the information being collected, processed and transmitted, possibly in real-time and in an automatic manner. The design of such a system has also to deal with the dynamism of the IoT environments and with the fact that the composition of input sources can evolve over time adding new sources or eliminating old ones. In this chapter, we present novel algorithms for evaluating the data security and quality levels of the data provided by different sources over time.

Such algorithms are integrated in an existing IoT middleware, named *NetwOrked Smart objects (NOS)* presented in our previous work [6] [7][8]. NOSs are computationally powerful devices connected to create a distributed processing and storage layer, able to process the data gathered from IoT data sources. NOSs collect the data generated by nearby IoT devices, process them and finally transmit the processed data on a publish/subscribe broker. Such a middleware includes functionalities for users and applications to dynamically specify their requirements in terms of data security and quality levels. In this way the architecture is able to assess data security and quality metadata and filter out only the data that satisfy users/applications needs. This adaptive behaviour represents a clear innovation over conventional one-size-fits-all approaches, which provide the same information to all consumers without considering their specific preferences. We also provide a prototype of the illustrated architecture in which real-time open data feeds are used.

The chapter is organized as follows: Section 2 describes literature contributions that consider security and data quality issues in IoT scenarios. Sections 3, 4, and 5 present, respectively, the architecture and the algorithms for evaluating security and data quality levels. The implemented prototype is instead described in Section 6, along with the results obtained by the validation phase. Section 7 ends the paper.

2 Related Work

One of the main factors limiting the growth and take-up of IoT is the lack of a reference model [9]. There have been many projects that tried to design a common architecture based on the analysis of the needs in this scenario. For example, an architectural reference model for the interoperability of IoT systems has been the main goal of the IoT-A (Internet of Things Architecture) Project [10]. A dynamic architecture for services orchestration and self-adaptation has been proposed in IoT.EST (Internet of Things Environment for Service Creation and Testing) [11]. The project defines a dynamic service creation environment that gathers and exploits data from sensors and actuators making use of different communication technologies and formats. Such an architecture deals with issues such as the composition of business services based on re-usable IoT service components, the automated configuration and testing of services for “things” and the abstraction of the heterogeneity of underlying technologies to ensure interoperability. Also the FP7 COMPOSE (Collaborative Open Market to Place Objects at your Service) project [12] focuses on composition. It aims to design and develop an open marketplace for IoT data and services. The basic concept underpinning such an approach is to treat smart objects as services, which can be managed using standard service-oriented computing approaches and can be dynamically composed to provide value-added applications to end users.

Another architecture has been proposed in the Ebbits project [13], that designed a SOA platform based on open protocols and middleware, effectively transforming IoT subsystems or devices into web services with semantic resolution. The goal was to allow businesses to integrate IoT into mainstream enterprise systems and to support interoperable end-to-end business applications.

Considering all these contributions, it is possible to state that researchers only agree that the basic model is a 3-layer architecture composed of Application, Network, and Perception layers [9] [14]. However, in the literature, several contributions propose other models that add more abstraction to the IoT architecture [14] [15].

In our work, we also tried to add different abstraction layers for designing a lightweight and flexible middleware for IoT applications [8]. In this chapter we aim to highlight the novelty and relevance if this solution, since it is the unique architecture that provides a comprehensive approach for managing data gathered from heterogeneous sources both addressing security and data quality issues. In fact, in the literature security and data quality issues have been addressed, but separately, as discussed in the next sections.

2.1 Security issues in IoT

In some proposal, security issues have been addressed. Typically, they focus on data communications: they enforce data to be exchanged according to strict protection constraints considering the heterogeneity of devices and communication technologies. Indeed, devices can be characterized by different technologies; for example, many smart devices can natively support IPv6 communications [16] [17]; while other existing deployments might not support the IP protocol within the local area scope and this requires the design of ad hoc gateways and middlewares [18]. Relevant contributions on security oriented IoT middlewares include: VIRTUS [19], which relies on the open eXtensible Messaging and Presence Protocol (XMPP) to provide secure event-driven communications; Otsopack [20] and Naming, Addressing and Profile Server (NAPS) [21], which are data-centric frameworks based on the usage of HTTP and REpresentational State Transfer (REST) interfaces.

Security aspects were also the central points of projects such as uTRUSTit [22] and Butler [23]. The approach pursued in the former one is to directly integrate the user in the trust chain, guaranteeing transparency in the underlying IoT security and reliability properties. If successful, the uTRUSTit approach shall enable system manufacturers and system integrators to express the underlying security concepts to users in a comprehensible way, allowing them to reason on the trustworthiness of such systems. Butler aims to allow users to manage their distributed profile by allowing data duplication and identity control over different applications. The final purpose is to deliver a framework able to dynamically integrate user data (e.g., location, behaviour) in privacy and security protocols.

2.2 Data quality issues in IoT

The huge number of data sources in IoT is considered a positive aspect for data fusion and for the extraction and provisioning of advanced services. However, it is important to exploit the benefits provided by the quantity of data only if a certain quality is guaranteed. Several literature contributions recognize data quality as one instrumental issue in IoT research. A survey was recently dedicated to this topic, stating that data quality is the basis for sound decisions [24]. In the survey, authors consider work that addressed quality issues in data streams and RFID data. They show that the main data quality dimensions are accuracy, confidence completeness, data volume and timeliness.

Other additional data quality dimensions to consider are mostly related to the infrastructure through which data are offered; in fact, they include the ease of access, the access security and the availability. Also in [25] quality is cited as a key parameter for an efficient access and use of IoT data and services, and [26] claims the need of control over data sources to ensure their validity, information accuracy and credibility. Other literature contributions are focused on specific data quality aspects and issues. [27] addresses only data accuracy timeliness and the trustworthiness of the data provider. In particular, authors propose anomaly detection techniques to remove noise and inaccurate data in order to improve data quality.

New data quality dimensions have been introduced in [28]. In fact, authors focus on uncertainty, redundancy, ambiguity and inconsistency. Such dimensions are mainly related to the fact that, in an IoT environment, data may be gathered from different sources and they can be characterized by different precision or accuracy or they can monitor the same phenomenon reporting duplicates or inconsistent values. All these issues have a negative effect on the source reliability.

3 Networked Smart objects Architecture

The architecture called NOS, proposed in [8], is illustrated in Fig. 1. It provides interfaces for the interaction with the sources and with the users. In fact, on the one hand, the architecture gathers input data from different kinds of sources (the so-called *E-Nodes*) that are represented by heterogeneous devices (e.g., wireless sensor networks, RFID, NFC, actuators, social networks); on the other hand, it lets the users access the offered IoT-based services through Internet-connected mobile devices (e.g., smartphone, tablet). In the following we briefly describe how the architecture works.

Starting from the sources, the architecture provides a service for the source registration by means of HTTP protocol. Registered sources are associated with an identifier, and, optionally, with a geographical position and/or an encryption scheme, including the proper keys for interactions with NOSs. For each incoming data NOSs extract the following information: (i) the data source, which describes the type of node (i.e., the identifier in case of a registered source); (ii) the communication mode, that is the way in which the data is transmitted (e.g., discrete or streaming communication); (iii) the data schema, which represents the type (e.g., number, text) and the format of the transmitted data; (iv) the metadata describing the data content; (v) a timestamp describing when the data were received by NOS. The HTTP communication protocol is also used among NOSs and the data sources for the data transmission. Since received data are highly heterogeneous, they are stored in the *Raw Data* repository, and, periodically, they are processed by the *Data Normalization* and *Analyzers* modules. The former puts the data in the format specified in Fig. 2. The latter periodically extracts the normalized data and computes the relevant security and data quality indicators (the related algorithms are described in Sections 4 and 5).

The processed data are used for providing services to the target users. The user interface is based on the Message Queue Telemetry Transport (MQTT) protocol, that

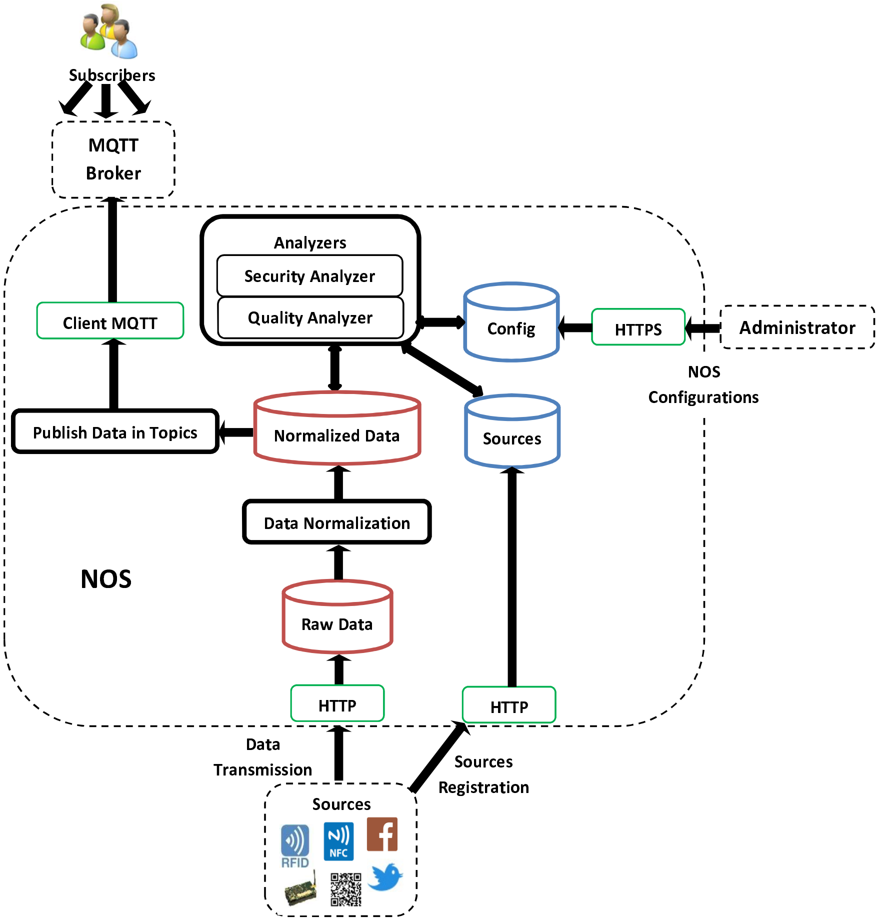


Fig. 1 NOS Architecture

is a lightweight publish/subscribe protocol [29] specifically designed for resource-constrained devices. The MQTT client exchanges messages with the MQTT broker by means of publications and subscriptions to topics. Such a mechanism is adopted to support interactions among services and IoT devices. The architecture includes a module in charge of assigning data items to the corresponding topic and to publish them on a MQTT broker, as depicted in Fig. 1. For this task, a taxonomy is needed: for example for publishing temperature information of a sensor with identifier *sensorId* could be *sensor/temperature/sensorId*. Note that subscribers may register for specific topics at runtime and our architecture provides a mechanism for dynamic subscription and unsubscription to topics. According to the MQTT protocol, messages can be published with a Quality of Service (QoS) parameter indicating that a message should be delivered "at most once", "at least once" and "exactly once". MQTT also supports persistence of messages to be delivered to future clients that subscribe to a topic, and may be configured to send messages of specific topics when the subscriber connection is abruptly closed. These parameters are specified in the *Config* storage unit. Summarizing, a typical MQTT message includes the following parameters: (i) the topic; (ii) the data value; (iii) the QoS level; (iv) the retain value.

4 Data quality evaluation

IoT architectures are designed to handle streams of data gathered from millions of intelligent devices; therefore, new challenges in data quality assessment raise. In fact, data volume, velocity and variety need to be properly addressed. The first two issues can be handled by considering a window-based approach for which the data quality of the data streams results from the periodic assessment of the set of values included in the different time windows. Different windows can be rapidly analysed by using parallel distributed processing (e.g., map-reduce approaches). Data variety requires adaptive mechanisms, able to activate the appropriate data quality dimension and the related assessment metric. For example, in case of numbers (e.g., temperature values gathered from a specific sensor or provided by a sensor), accuracy and precision have to be calculated, but in case of text (e.g., tweets) the source reputation could be more relevant than the evaluation of a syntactic accuracy. However,

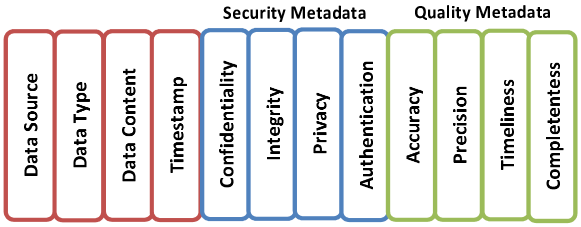


Fig. 2 NOS data format

all the data quality metadata assessed for the different windows are aggregated to define the source data quality level. In particular, in our approach, the data quality metadata have been defined as scores in the range $[0,1]$. For the case study defined in Section 6, timeliness, completeness, accuracy and precision levels have been considered [30] [31] and evaluated by the *Quality Analyzer* (see Figure 1).

Timeliness is defined as the temporal validity of data and is calculated on the basis of the freshness of data and on the frequency of data updates. The former is called *Currency* and defines the interval from the time when the value was sampled to the time instant at which data are received by NOSs. The latter is called *Volatility* and indicates the amount of time units (e.g., seconds) during which a data remains valid; it is usually associated with the type of phenomena that the system has to monitor and depends on the timescale of its dynamics.

Completeness provides information about the quantity of values received by the source. In particular, it analyzes if the data set includes all the values that a sensor or device was supposed to gather. In fact, it is calculated as the amount of collected values in a given time interval with respect to the amount of expected values. Completeness is important since the percentage of missing values can be a good indicator for revealing sensors inefficiencies or communication issues.

Accuracy is usually defined as correctness and it is measured as the degree of similarity between the value stored in the system and the right value. More formally, the accuracy is based on the evaluation of the error ε_{acc} resulting from the difference between the sensed value v_n and a reference value v_{ref}. Accuracy is usually related to *Precision*, conceived as the degree to which further measurements of the same phenomenon in a close time instant provides the same or similar results. Precision is often specified in terms of the standard deviation of the measured values: the smaller the standard deviation, the higher the precision. A correct representation is characterised by accurate and precise values. However, in continuous quality monitoring, changes in accuracy and precision can reveal errors or changes in the monitored process. In particular, precise but inaccurate values can be caused by changes in the monitored phenomenon or by faulty sensors [30].

5 Security evaluation

Together with data quality metadata, security metadata are exploited to understand the nature and reliability of the sources managed by the IoT platform. The *Security Analyzer* is responsible for the security assessment and must be able to access to the *Sources* storage unit in order to analyze the received data in relation to the source that sent them to NOSs.

In more detail, NOSs associate a score in the range $[0,1]$ to each security metric. As in the IoT context sensitive data are often managed, the security scores are intended as levels of *confidentiality* and *integrity* of the received information, *privacy* of the transmitting source and *authentication* (i.e., the robustness of the source authentication). Note that malicious devices may be represented by non-registered sources, which send violated data to the IoT platform or execute malicious actions towards those transmitted by non-malicious ones (e.g., spoofing, sniffing).

In order to assess security, it is necessary to consider two groups of elements: a set A of threats/attacks a_n and a set C of security countermeasures c_m. The former includes the attacks that may impact on the data managed by the platform (e.g., data violation, unauthorized access, masking, impersonation). The latter regards the countermeasures available in the platform to face the attacks (i.e., encryption, authentication, key pre-distribution). The security model considered by this algorithm links the attacks of a_n with the corresponding countermeasures in c_m. The taxonomy of the security attacks and the related countermeasures is retrieved from [32], which also considers that each countermeasure is characterized by a degree of resistance to a violation or to an attack attempt. On such a basis, we are able to define the relationships among attacks and countermeasures considering that an attack can be tackled through a plurality of countermeasures and a countermeasure can face more than one attack. Each relationship is associated with a weight w_{a_n,c_m} in the range $[0:1]$, which represents the level of robustness of the countermeasure c_m with respect to the attack a_n (see Figure 3).

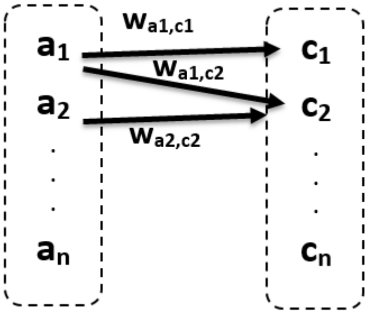


Fig. 3 Weighed relationships among attacks and countermeasures

The identified relationships can be classified on the basis of the considered security metrics in a way to obtain four groups that might also overlap: (i) g_{conf} for attacks-countermeasures related to the data confidentiality; (ii) g_{int} for the pairs related to data integrity; (iii) g_{pri} for privacy issues; (iv) g_{auth} for the pairs concerning sources' authentication. Note that such a model has to be defined at design time, stored in the collection named *Config*, and can be updated at runtime when new attacks and/or countermeasures have to be considered.

Table 1 shows some examples of attack-countermeasure pairs derived from the used taxonomy and classified in the groups described above.

The weights associated with the relationships are at the beginning set to 1. They might be updated at runtime on the basis of the malicious events that occur in the IoT system (detected by a monitoring system installed on NOSs). Weights can thus vary over time in a dynamic way; such a process of automatic adjustment is performed

Table 1 pairs attack-countermeasure

Attack	*Countermeasure*	*Group*
1) Packet sniffing	Data content encryption	g_{conf}
2) Password attack	Complex password generation	g_{conf}, g_{auth}
3) Man-in-the-middle attack	Data content encryption	g_{int}
4) Session hijacking attack	Secure session establishment	g_{int}, g_{auth}
5) Identity spoofing	Identity encryption	g_{pri}
6) Key impairment	Secure key distribution scheme	g_{conf}, g_{auth}

by means of a well-known learning approach, namely difference temporal learning [33]. Please refer to [8] for further details on this method.

Once the attacks/countermeasures model is defined, the algorithm computes for each incoming data the related security scores on the basis of the actual weights and of the data source s_k. A general equation for the assessment of the different security dimensions sec_dim (i.e., confidentiality (sec_{conf}), integrity (sec_{int}), privacy (sec_{pri}) and authentication (sec_{auth})) can be defined as:

$$sec_{dim} = \frac{a_{dim,s_k}}{a_{g_{dim}}} \cdot \frac{\sum i \varepsilon a_{dim,sk}, j \varepsilon c_{dim,s_k} w_{i,j}}{c_{dim,s_k}} \quad (1)$$

where: $a_{dim,src}$ is the number of attacks related to a specific security dimension that the source s_k could suffer; $a_{g_{dim}}$ is the total number of attacks included in the group g_{dim} (valid for any type of sources); c_{dim,s_k} is the number of countermeasures adopted by s_k related to the attacks to the specific security dimension included in a_{dim,s_k}. The sum of the weights considers only the weights between the attacks in a_{dim,s_k} and the countermeasures in c_{sim,s_k}.

For example, let us consider confidentiality, and let us suppose that the source s_k adopts AES for encrypting its data; moreover, it also adopts a 8-bit length password as credential for ensuring both confidentiality and authentication. As shown in Table 1 (points 1 and 2), AES is a countermeasure associated to the g_{conf} group; while the password is associated to both g_{conf} and g_{auth} groups. The steps performed by NOS to assess over the time the confidentiality score sec_{conf} are the following:

- The initial weights corresponding to the two pairs attack-countermeasure (i.e., AES-packet sniffing, 8-bit password-credential violation) are set to 1 and the first confidentiality score sec_{conf} is evaluated
- During the system operations, the platform recognizes no violated packets from the source s_k, but several times its password has been intercepted (e.g, through brute-force attack). As a consequence, the weight related to the pair 8-bit password-credential violation decreases; for such an example, let us assume that it is updated to 0.3 by the learning algorithm
- The new data obtained from the source s_k will receive a lower confidentiality score sec_{conf}, which is recomputed to 0.65.

As a consequence, a user who wants to receive data from the source s_k will be aware that they have a level of confidentiality not greater than sec_{conf}, so there is a $((1 - sec_{conf}) * 100)\%$ risk of a confidentiality attack.

Knowledge and metadata required to properly assess data quality and security levels are stored in a proper format in the repository *Config*. Such a unit contains all the configuration parameters required for the correct management of the IoT system (e.g., how to calculate quality properties on the basis of the data type, which attacks or security countermeasures to consider), represented in JSON format (as described in Section 6). Therefore,*Analyzers* periodically query the *Config* storage unit in order to know which rules shall be used.

6 Prototype and validation

The NOS system presented in Section 3 has been implemented [1] by using the following technologies: (i) the *Node.JS* platform[2] has been used for the platform implementation; (ii) *MongoDB*[3] for storage management; (iii) Mosquitto[4] for the publish/subscribe system. Modules interact among themselves via *RESTful* services.

We deployed a prototypical service middleware platform able to manage a large amount of data from heterogeneous devices with lightweight modules and interfaces working in a non-blocking manner to perform data analysis, discovery, and query [8]. Such a platform is innovative for different aspects. First of all, one or more NOSs can be deployed in a distributed manner without using a peer-to-peer management, since they are completely independent from each other. This is a novel approach with respect to the conventional ad hoc centralized IoT solutions. Moreover, such solutions are often hardly reconfigurable [34], because they are conceived for very specific applications, based on a vertical silo-based approach. The middleware presented in this work supports, instead, dynamic reconfiguration and can be remotely orchestrated through Internet/intranet protocols, which are based on open standards (see Section 3).

As just said, a great advantage of this approach is that changes in the platform can be performed in a non-blocking manner: it is possible to introduce new modules, duplicate the existing ones or remove them without re-starting the whole system. Furthermore,the use of the non-relational *MongoDB* database allows the platform to be flexible since the data model can dynamically evolve. Finally, we obtain good performance in data access, especially in read/write operations using the in-memory capability of *MongoDB*: the IoT-generated data contained in *Raw Data* and *Normalized Data* repositories are not persistent; only the databases *Config* and *Sources* are persistent in the platform.

[1] The code is released as open source under a permissive license `https://bitbucket.org/alessandrarizzardi/nos`

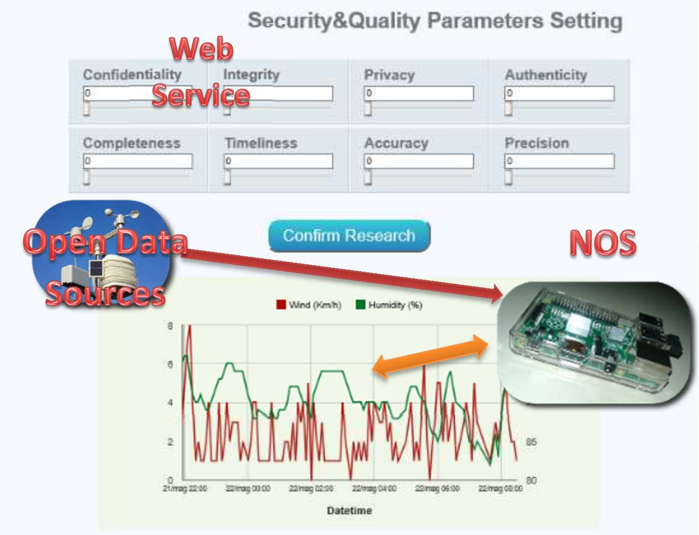
[2] `http://nodejs.org/`

[3] `http://www.mongodb.org/`

[4] `http://mosquitto.org`

We tested the NOS platform by deploying it on a Raspberry Pi, and connecting it to a number of open data feeds. In particular, we exploited six sensors at the meteorological station in the city of Campodenno (Trentino, Italy) that provide real time data referred to temperature, humidity, wind speed, energy consumption and air quality. Data are transmitted by a web service that exposes them in *JSON* format, and the NOS retrieves them through HTTP *GET* requests. According to the system presented in Section 3, data are analyzed from a security and quality perspective following the methods presented in Sections 4 and 5 and then transmitted to the MQTT broker.

Through a simple visualization service users can set their preferences in terms of security, privacy and data quality and access to the metadata calculated for the incoming values. The dashboard is shown in Figure 4.



Security&Quality Parameters Setting

Confidentiality 0	Integrity 0	Privacy 0	Authenticity 0
Completeness 0	Timeliness 0	Accuracy 0	Precision 0

Fig. 4 User Dashboard

For testing the effectiveness of the proposed mechanisms, the system has been observed for a period of a week. Results have been discussed in [8]. We want only to highlight that, as specified in Sections 4 and 5, each score is initially set to the maximum value (i.e., 1); values are then updated following changes in security and quality aspects. From the figures, it can be observed that some sources are characterized by a good level of authentication, but, at the same time, by a low level of reliability in terms of confidentiality, integrity and privacy (e.g., source 6 in Figure 5) and viceversa (e.g., source 3 in Figure 5). From the data quality perspective, in some sources, the provided data present good levels of completeness and timeliness, but poor accuracy and precision levels (e.g., source 4 in Figure 6).

The experiments confirmed the usefulness of the presented approach in empowering the users to retrieve only the data that meet their requirements.

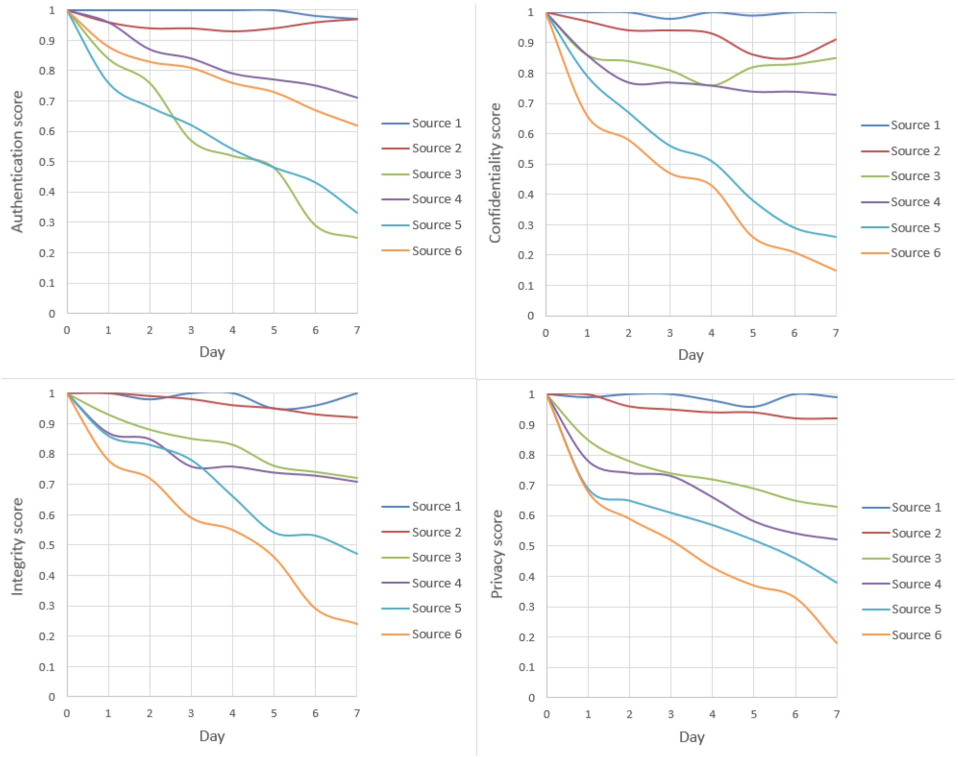


Fig. 5 Security Score Evaluation

Other useful metrics about NOSs' performance, in terms of overhead, memory occupancy, computational load and latency, have been evaluated in other recent work, which include: (i) the integration with an enforcement framework [35], conceived for guaranteeing a fine-grained access control against violation attempts; moreover, it allows to enforce policies specifically related to the security and data quality levels presented in this chapter; (ii) a protocol, named AUPS (AUthenticated Publish/Subscribe), for the authentication of the communications taking place via MQTT [36]; it is able to manage the disclosure of data on the basis of the policies associated to the defined topics; (iii) the integration of two key management systems, originally conceived for wireless sensor networks and adapted to the NOS platform [37]; they provide NOSs with the capabilities of handling the distribution and the replacement of the encryption keys among users and data sources. In all these cases, NOS platform demonstrated a good trade-off between a correct behavior and efficiency.

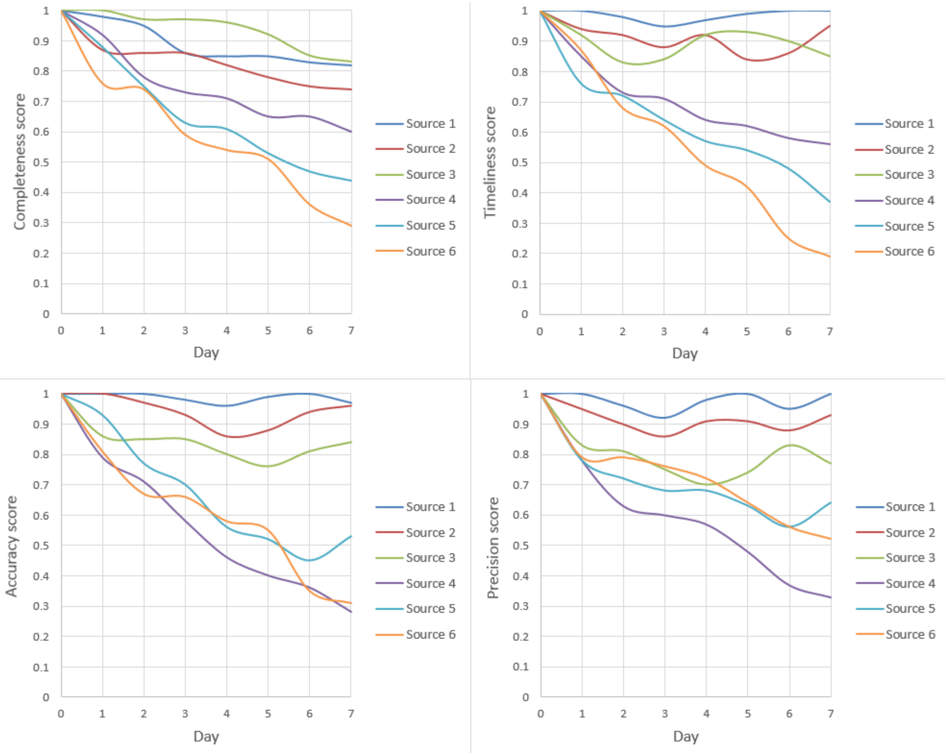


Fig. 6 Quality Score Evaluation

7 Conclusions

In this chapter we have presented the design and a prototypical implementation of a distributed IoT middleware layer, named NOS, able to manage heterogeneous data sources, to provide a uniform, consistent data representation and to provide data services to manage and filter data on the basis of their related security and data quality metadata. Such an architecture improves data governance in IoT environments, since, on the one hand, it manages efficiently data and, on the other hand, it also addresses security and data quality issues by improving the user's awareness about the reliability of the accessed data. In this way, it is possible to provide data that are fit for the use in a specific context/application and thus valuable results.

The effectiveness of the proposed solution has been validated through the implementation of a real prototype of the NOS platform. Future work will focus on the design and the development of new methods for assessing further data quality dimensions and for dealing with other types of sources. Moreover, new sources will be considered for the evaluation of the proposed architecture.

Acknowledgements This work is partially funded by the Department of Theoretical and Applied Sciences of University of Insubria (Varese, Italy).

References

1. D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of Things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.
2. S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.
3. A. Coen Porisini, P. Colombo, and S. Sicari, *Privacy aware systems: from models to patterns*, igi global ed. Software Engineering for Secure Systems: Industrial and Research Perspectives, 2011.
4. L. Berti-Equille and J. Borge-Holthoefer, *Veracity of Data: From Truth Discovery Computation Algorithms to Models of Misinformation Dynamics*, ser. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2015. [Online]. Available: http://dx.doi.org/10.2200/S00676ED1V01Y201509DTM042
5. S. Sicari, A. Rizzardi, C. Cappiello, and A. Coen-Porisini, "A NFP model for internet of things applications," in *Proc. of IEEE WiMob*, Larnaca, Cyprus, Oct 2014, pp. 164–171.
6. S. Sicari, C. Cappiello, F. D. Pellegrini, D. Miorandi, and A. Coen-Porisini, "A security-and quality-aware system architecture for Internet of Things," *Information Systems Frontiers*, pp. 1–13, 2014.
7. A. Rizzardi, D. Miorandi, S. Sicari, C. Cappiello, and A. Coen-Porisini, "Networked smart objects: Moving data processing closer to the source," in *2nd EAI International Conference on IoT as a Service*, Oct 2015.
8. S. Sicari, A. Rizzardi, D. Miorandi, C. Cappiello, and A. Coen-Porisini, "A secure and quality-aware prototypical architecture for the internet of things," *Inf. Syst.*, vol. 58, pp. 43–55, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.is.2016.02.003
9. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, Fourthquarter 2015.
10. "IOT-A project," http://www.iot-a.eu/.
11. "IOT-EST project," http://ict-iotest.eu/iotest/.
12. "European FP7 IoT@Work project," http://iot-at-work.eu.
13. "EBBITS project," http://www.ebbits-project.eu/.
14. Z. Yang, Y. Yue, Y. Yang, Y. Peng, X. Wang, and W. Liu, "Study and application on the architecture and key technologies for iot," in *2011 International Conference on Multimedia Technology*, July 2011, pp. 747–751.
15. M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for internet of things: A survey," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 70–95, Feb 2016.
16. M. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. Grieco, G. Boggia, and M. Dohler, "Standardized protocol stack for the Internet of (important) Things," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 3, pp. 1389–1406, Third 2013.
17. I. Bagci, S. Raza, T. Chung, U. Roedig, and T. Voigt, "Combined secure storage and communication for the Internet of Things," in *2013 IEEE International Conference on Sensing, Communications and Networking, SECON 2013*, New Orleans, LA, United States, June 2013, pp. 523–631.
18. D. Boswarthick, O. Elloumi, and O. Hersent, *M2M Communications: A Systems Approach*, 1st ed. Wiley Publishing, 2012.
19. D. Conzon, T. Bolognesi, P. Brizzi, A. Lotito, R. Tomasi, and M. Spirito, "The VIRTUS middleware: An XMPP based architecture for secure IoT communications," in *2012 21st International Conference on Computer Communications and Networks, ICCCN 2012*, Munich, Germany, July 2012, pp. 1–6.

20. A. Gòmez-Goiri, P. Orduna, J. Diego, and D. L. de Ipina, "Otsopack: Lightweight semantic framework for interoperable ambient intelligence applications," *Computers in Human Behavior*, vol. 30, pp. 460–467, January 2014.
21. C. H. Liu, B. Yang, and T. Liu, "Efficient naming, addressing and profile services in Internet-of-Things sensory environments," *Ad Hoc Networks*, vol. 18, no. 0, pp. 85–101, 2013.
22. "Usable trust in the Internet of Things," http://www.utrustit.eu/.
23. "BUTLER project," http://www.iot-butler.eu.
24. A. Karkouch, H. Mousannif, H. A. Moatassime, and T. Noel, "Data quality in internet of things: A state-of-the-art survey," *Journal of Network and Computer Applications*, vol. 73, pp. 57 – 81, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804516301564
25. P. Barnaghi and A. Sheth, "On searching the internet of things: Requirements and challenges," *IEEE Intelligent Systems*, vol. 31, no. 6, pp. 71–75, Nov 2016.
26. B. Guo, D. Zhang, Z. Wang, Z. Yu, and X. Zhou, "Opportunistic iot: Exploring the harmonious interaction between human and the internet of things," *J. Netw. Comput. Appl.*, vol. 36, no. 6, pp. 1531–1539, Nov. 2013.
27. A. Metzger, C.-H. Chi, Y. Engel, and A. Marconi, "Research challenges on online service quality prediction for proactive adaptation," in *Software Services and Systems Research - Results and Challenges (S-Cube), 2012 Workshop on European*, June 2012, pp. 51–57.
28. Y. Qin, Q. Z. Sheng, N. J. Falkner, S. Dustdar, H. Wang, and A. V. Vasilakos, "When things matter: A survey on data-centric internet of things," *Journal of Network and Computer Applications*, vol. 64, pp. 137 – 153, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804516000606
29. "IBM and eurotech, "mqtt v3.1 protocol specification"," http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/ mqtt-v3r1.html.
30. C. Cappiello and F. A. Schreiber, "Quality- and energy-aware data compression by aggregation in WSN data streams," in *Proceedings of the 2009 IEEE International Conference on Pervasive Computing and Communications*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 1–6.
31. A. Klein and W. Lehner, "Representing data quality in sensor data streaming environments," *J. Data and Information Quality*, vol. 1, no. 2, pp. 10:1–10:28, Sep. 2009.
32. T. Roosta, S. Shieh, and S. Sastry, "Taxonomy of security attacks in sensor networks and countermeasures," in *The first IEEE international conference on system integration and reliability improvements*, vol. 25, 2006, p. 94.
33. G. Tesauro, *Practical issues in temporal difference learning*. Springer, 1992.
34. I. Mashal, O. Alsaryrah, T.-Y. Chung, C.-Z. Yang, W.-H. Kuo, and D. P. Agrawal, "Choices for interaction with things on Internet and underlying issues," *Ad Hoc Networks*, vol. 28, no. 0, pp. 68–90, 2015.
35. S. Sicari, A. Rizzardi, D. Miorandi, C.Cappiello, and A. Coen-Porisini, "Security policy enforcement for networked smart objects," *Computer Networks*, vol. 108, pp. 133–147, 2016.
36. A. Rizzardi, S. Sicari, D. Miorandi, and A. Coen-Porisini, "Aups: An open source authenticated publish/subscribe system for the Internet of Things," *Information Systems*, vol. 62, pp. 29–41, 2016.
37. S. Sicari, A. Rizzardi, D. Miorandi, and A. Coen-Porisini, "Internet of Things: Security in the keys," in *12th ACM International Symposium on QoS and Security for Wireless and Mobile Networks*, Malta, Nov 2016, pp. 129–133.