Multimed Tools Appl
https://doi.org/10.1007/s11042-017-5431-1

Enhancing dependability through profiling in the collaborative internet of things

Imad Belkacem[1] · Safia Nait-Bahloul[1] · Damien Sauveron[2]

Received: 14 August 2017 / Revised: 1 November 2017 / Accepted: 14 November 2017

Abstract The future of the Internet of Things (IoT) is the Collaborative Internet of Things (C-IoT) in which different IoT deployments collaborate to provide better services. For instance, in smart city scenarios, C-IoT will have the potential to provide immersive multimedia user-experiences based on content and context fusion, immersive multi-sensory environments, location-based and media internet technologies, and augmented reality. However, this future paradigm will only be possible if the right decisions can be made based on the analysis of huge volumes of collected data: i.e. if the dependability of C-IoT is ensured. To address this challenge, we studied a simplified view of a C-IoT architecture composed of devices using three different technologies that have enabled the existence of IoT (RFID, NFC and Beacons). However, our proposal could be extended to any other devices in the context of C-IoT. To enhance the dependability of C-IoT, we deploy statistical data analysis techniques to improve the quality of the data obtained from identification and sensing devices and to select the most reliable devices that provide trusted (i.e. non-faulty) data in order to support accurate decision-making.

Keywords IoT · C-IoT · Collaboration · Dependability · Identification · Sensing · Data analysis

✉ Damien Sauveron
damien.sauveron@unilim.fr

Imad Belkacem
imad.belkacem@univ-mosta.dz

Safia Nait-Bahloul
Nait-bahloul.safia@univ-oran.dz

[1] LITIO Laboratory, University of Oran1, Ahmed Ben Bella, BP 1524, El-M'Naouer, Oran, Algeria

[2] XLIM (UMR CNRS 7252 / Université de Limoges), 123 avenue Albert Thomas,
87060 Limoges Cedex, France

Published online: 24 November 2017

1 Introduction

In 1999, Kevin Ashton coined the term “Internet of Things” [22], which is now used to describe the interconnected networking of physical devices, objects, appliances, sensors, or actuators collectively referred to as *things*. These devices can collect and exchange data over the network. Thus, today, in a smart home scenario, it is not uncommon for persons living in a house to have several connected objects such as smart lightbulbs, smart thermostats, heating systems, toothbrushes and fitness bracelets, a smart TV, and CCTV surveillance. These devices can communicate directly with each other or with more powerful servers located on the cloud to provide new services to users. According to current estimations, 200 billion devices will be connected by 2020 [17]. From the observation that IoT deployments were often segregated in specific domains (smart health, smart education, smart industry, etc.), Fawzi Behmann [3], introduced and developed the concept of the Collaborative Internet of Things (C-IoT) whose core aim is to leverage collaborative intelligence into existing and future application domains to provide enhanced services. Thus, instead of having separate applications for specific tasks such as a) smart city application managing and scheduling traffic signals, movements of vehicles, persons and animals, b) video surveillance systems, c) smart home and smart building systems, d) smart shops, e) smart health systems, and f) a smart grid), there will be cross-linkages between these applications, allowing collaboration that may significantly improve the quality of human existence and business efficiency in general.

1.1 Some ground-breaking technologies that enabled the IoT

Among the technologies that make IoT possible, obviously there are communication technologies, particularly wireless ones such as Wi-Fi, Bluetooth and ZigBee. But one in particular has made IoT a reality: Radio Frequency Identification (RFID).

RFID has always been considered a pivotal technology in the IoT since it allows the labelling of any physical object with an electronic tag to enable remote identification. The first RFID tags only sent their unique identifier (ID), normalized under the EPC (Electronic Product Code) system. Since then, RFID tags have been enriched with embedded sensors, enabling them to collect environmental data in real time and to store and share on demand with stakeholders of the ecosystem of assets to which they are attached. These sources of information created the first step toward the IoT.

At the same time, short range RFID technologies (i.e. those based on low frequency (LF) and high frequency (HF)) were developed to provide user-convenient security solutions for several industries, including building access control, payment systems and transport. In 2005, when the mobile phone was starting to become the digital Swiss Army Knife of many users, short-range HF RFID technology was normalized under the name of Near Field Communication (NFC) and made available in new smartphones. NFC created a revolution for ordinary cellphone users because it enabled user-convenient interactions between the digital world and the real world (for instance it was possible in a museum to get multimedia information on an artwork merely by scanning an NFC tag attached to the description of the artwork). Such a tag can connect a URL to an external resource that the smartphone can read. In some ways, this type of NFC tag is similar to the QR Code technology. However, NFC technology is more flexible since tags can store different types of information; also, their content can easily be updated with new information according to access rights policies defined by their owners. This is impossible with a QR Code. Moreover, as with RFID tags, NFC tags can contain embedded sensors to provide more contextual information to users.

A more recent promising technology for IoT is the Bluetooth Low Energy (BLE) beacon which, like NFC technology, enables a user to interact with the physical world by permitting localization operations. For instance, it can be used to determine the physical location of user in an indoor environment, to track assets, and even to trigger a location-based action on a device such as a check-in on social media or a push notification such as a video advertisement. As with RFID and NFC technologies, in smart beacons, sensors have been added to provide more contextual information.

What is interesting with the three ground-breaking technologies selected is that they were developed sequentially, first to establish the possibility of the IoT, and later to the point where they are now all integrated naturally in any IoT deployment. These technologies share some features such as identification, but they are complementary either because they operate in different spatial ranges for identification, or because they are associated with different multimedia-aware applications.

Altogether these three technologies, equipped with environmental and contextual sensors, have the potential to carry out collaborative work in C-IoT scenarios.

1.2 Dependability in C-IoT

As mentioned above, RFID, NFC or beacon-enabled devices with equipped sensors enable different geographically remote interconnected sites (e.g. warehouses, factories, shops, houses, buildings, hospitals, public spaces) to cooperate and share information collected from different nodes, to achieve a common objective. For instance, in [33], the authors explain in the context of smart cities how technologies like content and context fusion, immersive multi-sensory environments, location-based technologies and media internet technologies, along with augmented reality through a collaborative intelligence, will enable immersive multimedia experiences for citizens.

1.2.1 Why is dependability a requirement for C-IoT?

As stated in [6], C-IoT will lead to a significant increase in the amount of data collected and outliers are thus inevitable. So, to make the right decision, decision makers, in a broad sense (i.e. any application, service or person, using data from different sources), should base their assessments and analysis on reliable and trusted information. Indeed, if a decision were to be taken based on erroneous reported data it could end in disaster for a critical application.

For instance, in a global healthcare system, several actors (e.g. doctor, pharmacist) can cooperate for patients' benefit by remotely monitoring patients' health status (including blood pressure, heart rate and respiratory rate). In ambient assisted-living, health data can be reported via wearable sensors located on (or even in) the patient's body, and from multiple sensors deployed not only in their houses, but also in many locations in cities (where sensors are deployed for different purposes and not specifically dedicated to healthcare functions). If, based on wrong reported data, a doctor raises an alert to make an urgent visit to save the life of a person who has no problem, this is an issue but it is not critical. However, if the wrong data means the doctor does not raise an alert when there is a real safety problem for the patient, this is not an acceptable outcome. The consequence could be devastating since the collaborative healthcare system would have failed in its main aim, which is to ensure the wellbeing of the patient.

An example of a real platform suitable for C-IoT is illustrated in Fig. 1: the Freescale home health hub (HHH) reference platform [28]. It provides seamless connectivity with various wired and wireless health care devices (e.g. blood pressure monitors, pulse

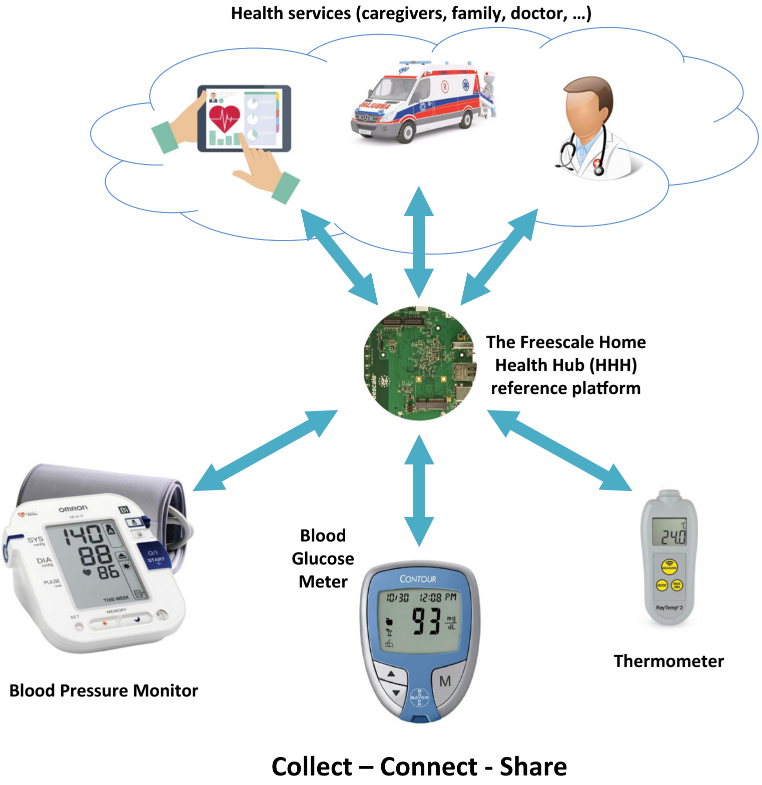
Health services (caregivers, family, doctor, ...)
The Freescale Home Health Hub (HHH) reference platform
Blood Pressure Monitor
Blood Glucose Meter
Thermometer
Collect – Connect - Share

Fig. 1 The Freescale Home Health Hub (HHH) platform

oximeters, blood glucose monitors, thermometers and weight scales). Collected data from these devices are then relayed to remote devices (e.g. smartphone, tablet or PC) in order to track and monitor the patient's health status and to provide alerts and medication reminders. The display interface can also provide a real-time connection to caregivers, including family, friends and physicians, to bring peace of mind and offer comfort and safety to the person being monitored.

In an application based on the platform in Fig. 1, data captured from several sensors are collected and securely stored in the cloud where they can be accessed by those involved in the patient's care and then shared through wireless connectivity with medical professionals who can make appropriate health recommendations. A concern is that making decisions about patients suffering from diabetes mellitus or hypoglycemia based on faulty glucose meter sensors to determine blood glucose levels is dangerous. A glucose meter is a key element in healthcare monitoring of these patients. The data collected in such a preventive approach may worsen the patient's health if decisions are based on erroneous information. For this purpose it is essential that reliable data is available.

According [3], another area in which C-IoT can be used is in home and building automation. As illustrated in Fig. 2, using smart devices like the Nest thermostat, smart bulbs and machine learning techniques to analyse users' habits and geolocations in conjunction with mobile applications can help to make users' environments respond seamlessly according to users' own preferences. Such systems can also help users to save money by switching on/off

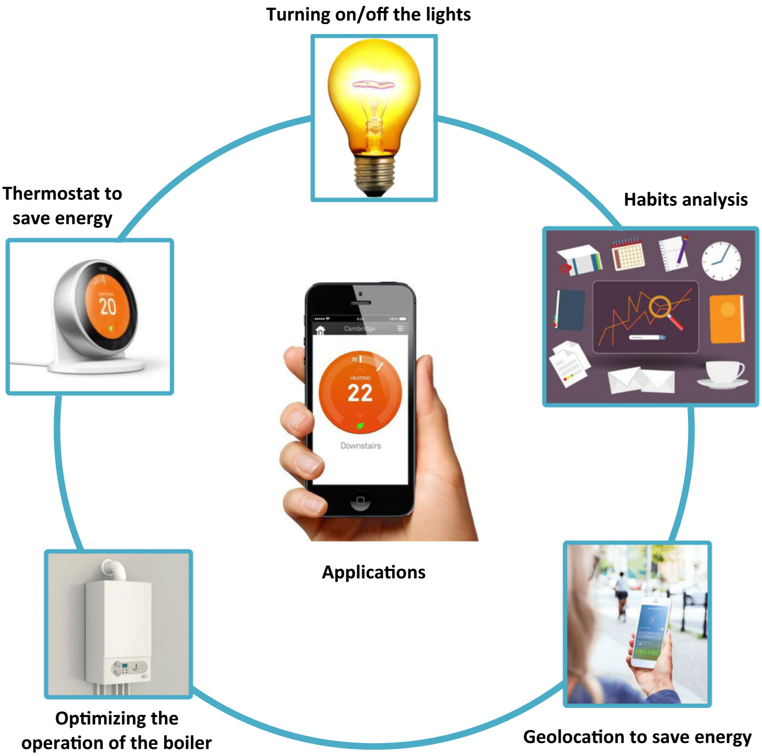
Turning on/off the lights
Thermostat to save energy
Habits analysis
Applications
Optimizing the operation of the boiler
Geolocation to save energy

Fig. 2 Example of an energy-managing scenario with C-IoT in the field of home and building automation

boilers and smart bulbs appropriately, while providing a technical framework that enables energy providers to use the collected information to more efficiently manage their smart grids to gain environmental benefits.

Thus, as explained above and illustrated in the healthcare scenario, the dependability of the C-IoT will be a major challenge. Although in current IoT deployment, designers of solutions are just beginning to address this issue, in this paper we offer a proposal to enhance dependability in C-IoT, based on our previous studies aimed at improving the dependability of RFID systems located in a single geographical site [4].

1.2.2 Dependability definition for C-IoT

Usually, dependability is defined as the property of a system such that we can justifiably place our reliance on the service it delivers [23]. In IoT, dependability is reliant on the networks that connect devices and on the dependability of the devices themselves (e.g. the quality and accuracy of measurements in the case of sensors). In C-IoT, in addition, dependability involves the architecture of mediation between the IoT systems.

1.2.3 Suitability of regular dependability mechanisms for C-IoT

Among the common methods to ensure local dependability, Local Fault Tolerance (LFT) systems can implement a voting mechanism that analyzes captured data and makes a correction if a fault is detected. Under the assumption that the voting mechanism is trustworthy, which is commonly accepted in the dependability community, the objective is thus to detect the fault in the system in which it is deployed. For example, in a Dual Modular Redundancy (DMR) system, two copies of the system are required and are expected to return similar results. Fault detection can be provided by simply comparing the results, but a voting mechanism cannot be used to correct the fault since it cannot decide which of the two systems is faulty. Thus, other corrective measures are needed to correct the fault. However, in a Triple Modular Redundancy (TMR) system, since three replicates are used and should provide the same data, a voting system would stop the propagation of the fault by detecting which replication is erroneous. Obviously, this is applicable if only one replication of the system is erroneous because if two replications are erroneous and provide the same faulty data, the voting system would regard the reliable replication as erroneous and the faulty replications as the reliable ones; the entire system would fail.

To resolve this issue and produce a more reliable system, the n modular redundancy (nMR) system has been proposed as a generalization of the TMR with n replications. However, while such an nMR system is very reliable, it is also expensive. In addition to the financial issues, there are many IoT deployments where an nMR-system could not be installed, either because of lack of availability of equipment or the complexity of management. Therefore, we do not believe an nMR system is an acceptable solution for the C-IoT.

1.3 Contributions

In this study, to enhance the dependability of the C-IoT, we use fault tolerance and statistical analysis of data to ensure the dependability of a studied C-IoT system composed of various geographically distributed remote sites in which devices using the three selected technologies (RFID, NFC and beacons), equipped with environmental and contextual sensors, are deployed. The physical aspects of data sensing and processing are beyond the scope of this paper.

We develop an approach to detect and correct faults that can occur, especially in sites where the LFT system is unreliable or where there is no LFT system. Reliability and correction of data for these sites are supported by a central server as follows:

- Collected data from the suspect device (e.g. RFID reader, sensor of a device) are compared with data from other similar devices in similarly reliable sites. If there is consistency between data, the device is considered reliable; otherwise the central server detects the fault and can help to correct it.
- Statistical hypothesis tests [36] on collected and corrected data are also performed at both the local and central level to support the process of making more reliable decisions about issues such as the occurrence of particular events and maintenance operation requirements.

Since particular geographical sites may not be able to maintain a permanent connection between the local server at the site and the central server, the proposed approach allows two communication modes while enhancing dependability. The first is the “always connected mode”, in which all devices maintain a persistent connection to the local server (and thus with the central server too) in order to make decisions based on events that may occur in real time. The second is the “serverless mode” which prioritizes local communication and for which a persistent connection to the remote central server is not permanently required [27]. In these less critical sites, the data analysis and the fault detection/correction processes occur periodically instead of in real time.

1.4 Structure of the paper

Section 2 presents the work carried out to ensure the dependability of the C-IoT. Section 3 gives the proposed architecture of a theoretical collaborative IoT and describes our methodology to improve its dependability. Section 4 details the statistical data analysis enabling the detection and correction of faults and supporting the accurate decision process. Section 5 concludes the paper and gives perspectives.

2 Related work

This section presents the related work on enhancing dependability in the C-IoT. We first review some studies aimed at improving reliability in RFID systems, which were the precursor of the IoT. Then we focus on selected studies related to the dependability of IoT and finally examine the recent work on the dependability of the C-IoT.

2.1 Dependability in RFID

As stated in [11], there are two methods, known as monitoring, to detect faults in RFID components during system operation. The first consists of a remote overall check that readers are suitably configured and functioning properly. The second method relies on observation of the overall performance of the readers according to several metrics: Average Tag Traffic Volume (ATTV), Read Errors to Total Reads (RETR) [38] and Read-ErrorRate (RERavg) [10]. Though these methods are effective, they only focus on the reader-tag relationship to ensure the dependability of the RFID system. Therefore, since modern RFID systems integrate several RFID readers and there is middleware to manage the whole system, it is possible to take advantage of it to ensure enhanced dependability at this level. In addition, since RFID tags equipped with environmental sensors are becoming important components of the RFID infrastructure to provide additional information to help in management of the tagged assets, it would make sense to collect, correct and merge data at the central level (i.e. the middleware) to improve the dependability of the system.

In [4], the authors used an online test method based on the confidence interval (CI) of the analyses performed at this central level. They compared the results of the inventories of the RFID readers and also compared the results of the homogeneous sensors in order to detect and correct faults that could affect the RFID system and to replace defective readers and sensors. However, this approach focused on a single geographical site whereas in the context of C-IoT, systems are more complex and involve several remote sites.

2.2 Dependability in IoT

To illustrate dependability research in the IoT area, in the RELYonIT (Research by Experimentation for Dependability on the Internet of Things) project [30], funded by the European Union, researchers have developed ways to significantly increase the reliability of IoT solutions faced with radio interference and temperature fluctuations. To study these phenomena, in [5], the authors designed low-cost extensions for wireless sensor networks testbeds to examine the impact of temperature (the TempLab testbed) and to create realistic interference models and reproducible interferences (the JamLab testbed).

In [24], the authors proposed mathematical models based on Markov chains, which were able to estimate the reliability and availability of IoT applications by considering redundancy aspects. They also proposed a set of redundancy models able to predict the reliability of IoT devices/applications.

In [8], by applying the principle of "dependability by design", the authors built a framework based on the concept of virtualized IoT services that support a variety of security models and implement them according to the requirements of the considered application. However, they did not provide a recovery mechanism for failed operations. In addition, their work was particularly focused on the context of Wireless Sensor Networks (WSN).

2.3 Dependability in C-IoT

The study most closely related to dependability in the C-IoT is [6]. The authors proposed a simple and efficient approach to selecting and refining the data from reliable sensors using a collaborative filtering technique. Their proposal involved looking for reliable sensors among a list of sensors by comparing the reference results of an already known reliable sensor with the results of the sensor whose reliability is being checked, using the Pearson (r) correlation coefficient. However, if a strong correlation is observed between the data of a reference sensor and a suspect sensor, this does not necessarily mean that the suspect node is reliable. More than a single comparison must be performed to confirm the reliability of the suspect sensor. In addition, the authors focus only on sensors (mainly on temperature sensors), without addressing RFID, NFC or beacon technologies.

We strongly believe that the C-IoT requires a fault-tolerant architecture including a model that uses the behaviour of different nodes at different geographical sites to assess the reliability of the data collected by these nodes in order to detect and correct faults as soon as possible and thus stop their propagation. This model, based on a fault-tolerant architecture, is required to ensure the correctness of the decisions taken by the C-IoT system (particularly in critical cases). Thus, as in [6], in this paper, the result from the suspect node is compared with the result from a reliable node but using the double homogeneity test. In our work, the reliability of a node is estimated by considering the confidence rate computed after a significant number of iterations (not a single one as in [6]) by comparing the results of the different homogeneous nodes observing the same phenomena in the same time interval by performing local or central comparisons. If the double homogeneity test reveals a significant difference, we locally investigate whether the cause is explicable or whether it is due to a possible malfunction. In the latter case, a correction of data is made using the Extreme Studentized Deviate (ESD) test to exclude erroneous data from future computations. The confidence rating of the suspect node is decreased.

3 Architecture of a theoretical C-IoT

In this section, we present the architecture of a theoretical C-IoT composed of devices using the RFID, NFC and beacon technologies and equipped with sensors. This architecture is used as a blueprint to develop and illustrate our model of statistical data analysis to enhance dependability, but our method can be generalized to any possible architecture of the C-IoT.

3.1 Principles of the proposed method

Since ensuring dependability in the C-IoT is a complex task due to the heterogeneity that may exist at application level, as well as at the level of technologies used, our proposal exploits the collaboration of several remote geographical sites to enhance the reliability of all services provided by the C-IoT to users in profiling the behaviour of each identification node (RFID, NFC or beacons-enable reading devices) and each sensing node (environmental sensors embedded in RFIDs, NFC tags or beacons). This profiling process enables selection of the most reliable nodes, to detect and correct errors that may occur in a particular site where there is either no LFT or an unreliable LFT system. In addition, as explained in Section 4, a statistical hypothesis test [36] is used on corrected data to minimize errors in the decision process.

The profiling process is a comparative study of timestamped data collected from different sites/nodes with similar conditions: i.e. identification data (RFID inventories, NFC identifiers and collection of individual/objects notified by beacons) and measurements of environmental sensors. It is worth noting that physical aspects related to nodes (origin and nature of energy, reading distance, memory size, communication rates, frequency range, communication distance, etc.) do not need to be considered in our proposal.

Because the objective of our method is to enable the C-IoT system to make the right decision, we adopted a hybrid approach to take advantage of both centralized and distributed approaches. For the centralized approach, our method relies on a central server to diagnose

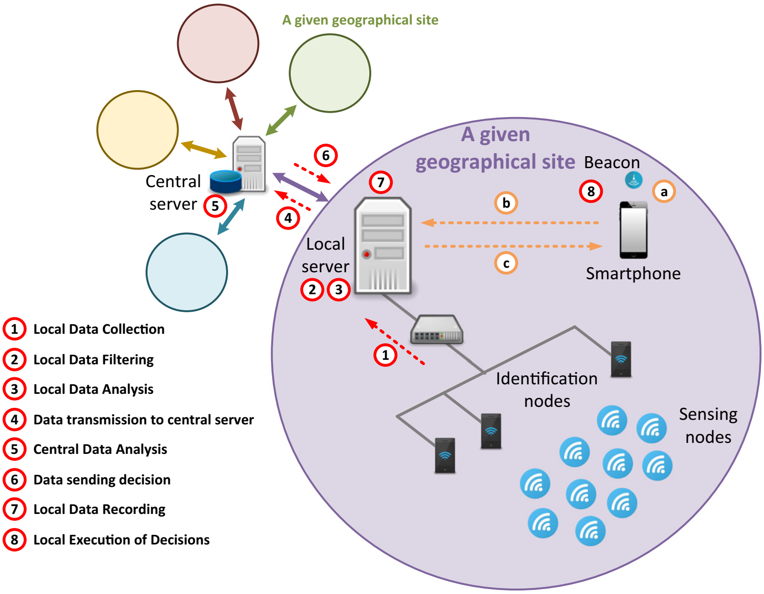

Fig. 3 Architecture of the considered C-IoT Note: steps 1,2,3,4,5,6,7 and 8 are related to the enhancing dependability process; steps a, b and c are related to push notification

faults based on the profiling process and on data gathered from the different geographical sites. This approach ensures a total monitoring of the status of the network to ensure accurate diagnostics for complex problems (e.g. critical node failures, occurrence of concurrent faults, or damage to a complete zone of nodes) [15]. To balance the limitations of this approach caused by the strong dependency generated by centralization, fault detection is also performed in a distributed manner by each site, to limit overloading of the main server.

3.2 Architecture overview

The architecture of the considered C-IoT is illustrated in Fig. 3. It is composed of several remote geographical sites which all communicate with a central server. Collaboration is ensured by the central server in two modes according to the criticality of the sites:

- The connected mode is used when the level of criticality is high and data analysis and fault correction must be done in real time to avoid important damage.
- The serverless mode is used when the risk is less important and data analysis and fault correction are done periodically (e.g. at the end of each day).

The central server manages several areas of general information about the geographical sites, including the degree of criticality, type of applications (e.g. transport, access control, or inventories), number of nodes installed and their technologies, and the IP addresses of local servers. Any change in local sites related to this information is updated on the central server since it needs them for its central analysis (see Section 4.2).

Every asset and person is tagged with RFID, NFC or beacon-enabled device (usually a tag) equipped with several environmental sensors. Each tag has its own unique identifier. Each site is managed by a local server running a database for the regular operations of identification of tagged assets or persons and of collection of environmental data sensed by embedded sensors on the tags. In a regular process, the identification nodes and sensing nodes, defined below, send their information to the local server periodically. However, this period can be shortened for critical sites.

Before providing details of the data analysis processes performed in the architecture, the three selected technologies are briefly described and examples of the considered devices, equipped with sensors for each technology, are presented in the rationale for our architecture of a C-IoT. In RFID technology, an RFID reader normally emits a radio-frequency signal to power the RFID tags, which then send the reader their identification (ID) and additional information by backscattering the RF field. There are essentially three frequency bands in which RFID tags operate. These frequencies are related to the distance of communication. Low frequency (125-134.2 kHz) and high frequency (13.56 MHz) enable communication over a few centimeters. Ultra-high frequency (860-960 MHz) enables communication over several meters. RFID tags can be equipped with environmental sensors (temperature,

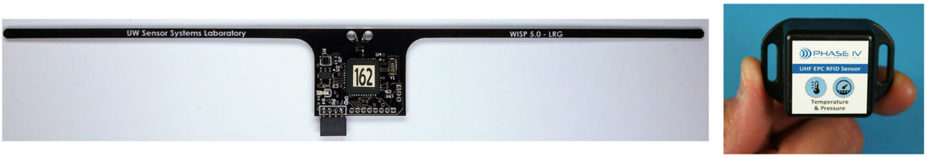

Fig. 4 Two passive UHF RFID tags: a WISP 5.1 tag [39] (on the left) and a PHASE IV Engineering tag (on the right - Photo courtesy of PHASE IV [18])

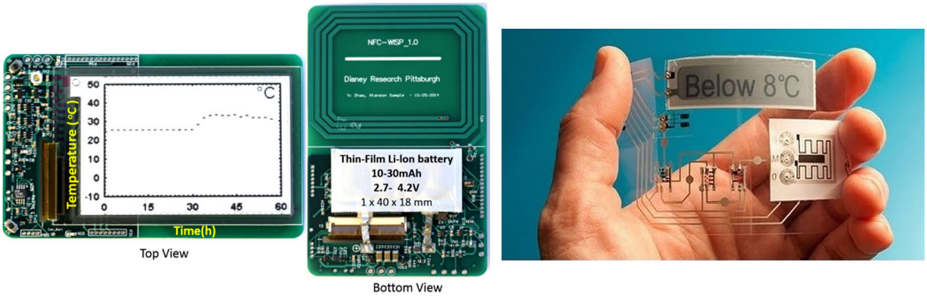

Fig. 5 Two NFC tags: a NFC-WISP 1.0 battery powered tag (on the left - Photo courtesy of NFC-WISP [32]) and a passive Thinfilm NFC tag (on the right - Photo courtesy of Thinfilm [37])

moisture, pressure, movement, etc.) to help to identify and to follow changes in the environment of the tagged assets. Examples of such devices are presented in Fig. 4. The WISP tag [39] is more for prototyping stages, while the PHASE IV Engineering tag [18] is used commercially to sense temperature, strain and pressure.

NFC is based on RFID technology operating in the HF (13.56 MHz) range. In reader mode, NFC involves two elements: an initiator and a target. The initiator (generally a reader or an NFC-enabled smartphone) actively generates an RF field that can power the passive target (generally a tag), which answers with its unique identity (UID) and additional information using a backscattered modulation technique [7]. NFC supports two additional modes of communication, called peer-to-peer mode and card-emulation mode. In peer-to-peer mode, ad hoc communication is possible, provided both devices are powered. In card-emulation mode, powered devices can act as passive NFC tags to communicate with other powered devices. As with RFID technology, there are NFC tags equipped with environmental sensors. These tags may be purely passive, semi-active (e.g. only sensors are powered) or active. Two examples are presented Fig. 5. The NFC-WISP [32, 40] is mainly used for prototype development, whereas the Thinfilm Company [37] product is commercially available.

Bluetooth beacons are powered devices that use Bluetooth Low Energy (BLE) technology (thus operating at 2.4 GHz) to broadcast a small amount of data to any BLE-enabled equipment (usually smartphones or tablets) in communication range (up to 100 meters). Initially, beacons were used to provide a more accurate indoor localization [16, 19, 20] than GPS. Today, this technology is used to send notifications to customers in shops, inform visitors in museums, guide people in stadiums, organize meetings, and to direct employees. The limitations of the first generation of beacons were related to the poor structure of the packets of information broadcasted, which contained 3 fields: a reference number (the so-called UUID or Universally Unique IDentifier), and two identification numbers, a major (for instance the ID of a shop) and a minor (for instance the ID of a shelf). These data were interpreted by an application installed on the device of the receiving party (installation of a dedicated application was required for each location using beacons). More recently, new kinds of beacons, like Google's Eddystone, have enabled the transmission of more information than the UIDD, like URI or data coming from environmental sensors (including sensors for acceleration, temperature, humidity, light and magnetism). Examples of beacons supporting sensors are the nRF51822 Bluetooth Smart Beacon [34] from Nordic Semiconductor [35] and the Asensor [1] from April Beacon company [2], shown in Fig. 6.

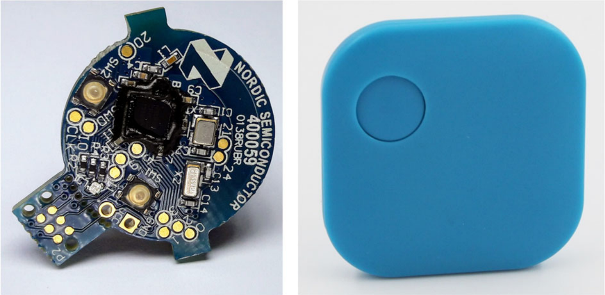

Fig. 6 Two beacons: an nRF51822 Bluetooth Smart Beacon without sensor attached (on the left) and an ASensor including a 3-axis, 12-bit accelerometer (on the right - Photo courtesy of April Beacon [2])

These three technologies are complementary to each other. They all have a unique identifier (ID, UID or UUID) but they may include different sensors depending on the environment of the tagged assets. They also have different communication ranges and modes of communication, allowing them to be used in a wide range of IoT applications. Costs vary markedly depending on the device under consideration: from very cheap for RFID (a few cents/tag) and NFC (1$/tag) to quite expensive for beacons (tens of dollars/beacon). These features mean that these technologies serve the IoT in different ways and they are used accordingly in the architectural blueprint designed to develop our approach.

In our considered architecture, there are two kinds of nodes.

- Identification nodes are RFID readers, NFC initiators or BLE-enabled devices (such as smartphones or tablets). When we discuss the reliability of the identification nodes for RFID or NFC technologies, it is related to the reliability of the RFID reader or of the NFC initiator, whereas for the beacon technology, it is related to the beacon device itself and the reading device. Behind the idea of identification nodes, there is the notion of inventories. For RFID and NFC technologies a device is required to read the passive tag to identify assets, whereas in beacon technology, since the beacon device is powered, errors in the inventory process can come from the beacon or from the reading device.
- Sensing nodes are environmental sensors embedded in RFID, NFC tags or beacons. There may be different sensing nodes on a same RFID, NFC tag or beacon. Data from these sensing nodes are read by identification nodes.

Later the notion of homogeneous nodes is used; therefore, a definition is needed:

- Two identification nodes are homogeneous when they use the same wireless communication technology.
- Two sensing nodes are homogeneous when they sense the same phenomena. For example, there may be two temperature sensors on different devices using different communication technologies (for instance one on an RFID tag and one on a beacon).

3.3 Overview of the process to enhance dependability of C-IoT

As explained above, in the considered architecture of C-IoT, at each site, there are n homogeneous nodes collecting particular data. If n is equal to 1 then the site has no LFT system for these particular data and thus if a fault occurs in the reading process, there will be no detection and therefore no local correction. If n is equal to 2, as discussed in Section 1.2.3, this site has, for these particular data, a DMR system (the minimal fault tolerance level for a system), which can detect the presence of a fault, but the correction must be based on other mechanisms than the voting system. A local fault correction can use a voting system when n is at least 3. In this case, n homogeneous nodes provide the same data and the majority voting system will enable determination of which replicate is wrong. So, when n is greater than 3, an nMR system can detect and correct the fault in a more efficient manner. Thus, in the following, if it is possible, a majority voting will take place at the local level (local server); otherwise the local site will request the help of the central level to analyze suspect data and use statistical comparison techniques to detect and correct erroneous values. Once notified that a decision process should be done, the local level can decide itself or can again solicit the help of the central server.

This section describes the main steps of the proposed process to enhance the dependability of the architecture of the considered C-IoT. The steps are described according the numbering in Fig. 3.

1. Local Data Collection: The local server collects data from various homogeneous nodes present at the site. Data are raw, i.e. not yet analyzed, but they can be exchanged in different formats (XML files, strings, etc.). The data includes information from RFID tags, NFC tags or beacon identifiers along with data collected by the environmental sensors.
2. Local Data Filtering: When data are received, the local server carries out transformations on the retrieved data streams and may exclude some data. Only the remaining data will be processed in the analysis phase.
3. Local Data Analysis: The local server analyzes the filtered data from the site in order to reach an appropriate decision (this is application-dependent and beyond the scope of this paper). If the site has no LFT or an inconclusive LFT system (for particular data) to enable it to trust the collected data, it will proceed to the next step to get support from the central server. It is worth noting that before requesting a central analysis, if a suspect result is obtained from local analysis, an operator should physically check it on any site without an LFT system, because the data may indicate the occurrence of a real event. For a fully automated process, i.e. without human verification, an LFT system (at least DMR or TMR) is required. However, it can be assumed that with the growth of the IoT, in C-IoT there will be an LFT system for any particular data at any given site.
4. Data transmission to the central server: Data are sent from the local server to the central server.
5. Central Data Analysis: The central server compares data from the different sites to detect and correct any faults. This phase is performed periodically. This analysis is one of the core contributions of this paper and is detailed in Section 4.
6. Data sending decision to local server: Corrected data and decisions are sent from the central server to the local servers at the relevant sites; i.e. the central server informs the local server which items of data it received should be excluded from future computations.
7. Local Data Recording: Received data and decisions are recorded in the local database.
8. Local Execution of Decisions: The decisions received are executed (as above, this is application-dependent and beyond the scope of this paper). In addition to the use of a beacon as a node of identification and production of data through its sensors in the considered C-IoT, the beacon may also be used to notify operators present at the local site to take corrective action (for instance, to replace a reader, or some tags) according to the following process:

a) Based on the beacon signal received, the operator's smartphone is localized.
b) An application on the operator's smartphone can send his location to the local server.
c) If the operator is located close to the defective equipment (node or tag), he will be notified by the local server with a push notification on his smartphone.

It is important to note that while our proposal can help to enhance the dependability of C-IoT, it does not solve all problems and administrators are still responsible for properly managing databases with backup processes, etc.

4 Analysis, correction and decision within the C-IoT

This section presents all the statistical tests used locally or centrally to ensure the dependability of the C-IoT. For fault detection, we propose the use of statistical tests of homogeneity. For fault correction, we propose the use of the generalized Extreme Studentized Deviate test (generalized ESD) to enable the server to discard any identified erroneous values in future computations. Based on the corrected data, the decision process uses statistical hypothesis tests [36] for an average and for a percentage to minimize the risk of error in making decisions. As stated in Section 3.1, all the proposed tests can be assimilated to a profiling process where suspect nodes or data are compared to references for detection, correction and decisions.

4.1 Local analysis

As in [4], a statistical analysis of data is performed by the local server at sites where an LFT mechanism exists. The following data are retrieved periodically:

- The vector of the inventories: $I = \begin{bmatrix} I_1 \dots I_n \end{bmatrix}$ where I_i is the number of tags/individuals/objects detected by the i^{th} identification node.
- The matrix of environmental data M (for each type of homogeneous sensors):

$$M = \begin{bmatrix} M_{11} & \dots & M_{1m} \\ \vdots & \ddots & \\ M_{n1} & & M_{nm} \end{bmatrix}$$

where M_{ij} is the observation of sensor S_j read by the i^{th} identification node.

Each line of the matrix M corresponds to the observations returned by a given sensor read to the different identification nodes.

Local analysis is based on these data (I and M) to detect defective hardware nodes (identification and sensing nodes, respectively). We use the confidence interval, CI, to detect faulty nodes based on the detected tags/objects/individuals vector for the identification nodes and the environmental observation matrix for the sensing nodes. The CI is an interval which contains, with a certain degree of confidence, the value to be estimated [21]. For instance, a confidence interval of 95%, (i.e. with a threshold of risk of 5%) has a probability equal to 0.95 of containing the value of the parameter. A CI is:

$$CI = \left]\overline{x} - \alpha\frac{\sigma}{\sqrt{n}}, \overline{x} + \alpha\frac{\sigma}{\sqrt{n}}\right[\qquad (1)$$

with σ: standard deviation, $\overline{x}$: mean, n: sample size, and α can be modified to have a wider or narrower interval. A larger α gives a wider interval which means fewer erroneous values detected and vice versa. After a learning phase consisting of a large number of tests, based on multiple iterations of the checking procedure to ensure that observations are within the CI before the system is in production mode, a confidence rate, CR, is computed for each node:

$$CR = \frac{\text{number of successful tests}}{\text{total number of tests}} \tag{2}$$

The CR is updated periodically based on the results returned by algorithm of Listing 1, described below. This rate is a measure of the reliability of the node since it reflects the quality of its results compared to the results of homogeneous nodes present at the same geographical site (using CI). If a significant difference in CR for some nodes is observed at a critical site, an intervention request is made using a push notification (explained in Section 3.3) to the operator concerned.

Since this mechanism is very efficient when the number of nodes (n) is large ($n > 3$), it can operate in a standalone manner. However, when $n \leq 3$, to get accurate decisions on the reliability of the nodes a comparison of results with homogeneous nodes present in other similar sites might be required, at a central level.

Algorithm 1 description For each geographical site in the considered C-IoT, Algorithm 1 is run periodically on the last fetch data for each type of homogeneous nodes (identification or sensors) present at the site. Since we propose a generic algorithm, as inputs, Algorithm 1 receives the type of node (i.e. `ident` for identification nodes or `sensor` for sensing nodes), the Inventory vector I for identification nodes or matrix M of capture data for sensing nodes and the confidence threshold `a`. The output of Algorithm 1 is the list of suspect nodes.

Once the list of suspect nodes is created, α in formula (1) is determined according to the number of homogeneous nodes present at the site. If this number is large enough, α is fixed for each CI; otherwise a consultation of the standard normal distribution table is necessary.

As already explained, if the degree of redundancy or the number of homogeneous nodes (n) is greater than 3, we have a reliable LFT system and the nodes whose values are outside the CI are suspected. For weak degrees of redundancy ($n = 2$ or $n = 3$), if the values are not equal, (or equal averages for sensors) and likewise for sites without LFT ($n = 1$), a central analysis will be requested to validate the data retrieved.

For suspect nodes, as stated in [4], data collected are invalidated and the mean of the valid homogeneous nodes at the same site is used.

Each time Algorithm 1 is run, the CR of the related nodes is updated.

4.2 Central analysis

As previously discussed, collaboration is carried out at the central level where the central server analyzes data collected from the different sites as requested.

Based on its knowledge of similar sites to the one containing suspect nodes (it is important to notice there can be more than one suspect nodes if the TMR system on the site requested the central analysis), the central server first performs a double homogeneity test to verify the behaviour of the suspect nodes.

Since the central server also records the CR (formula (2)) for each node that has been involved in the comparison process, when checking the behaviour of the suspect nodes, it

```
Inputs :  type /* Type of nodes - can be ident or sensor */
          I /* Inventory Vector if type=ident */ or M /* Capture matrix if type=sensor */
          a /* The confidence threshold */
Outputs : SN /* List of suspect nodes */
Begin
  Create_List(SN) /* Create an empty list of suspect nodes */

  /* Determine alpha for the formula 1 */
  If (n<30) Then
     alpha <- Student_distribution((1-a)/2,n-1)
  Else /* Large samples */ alpha <- u((1-a)/2) /* Consult the standard normal distribution table */
  EndIf

  If (type=ident) Then n <- size(I) Else /* type=sensor */ n <- number_of_lines(M) EndIf

  If (n>3) Then /* nMR Redundancy with n generally odd */
    If (type=ident) Then
       m <- mean(I)
      sd <- standard_deviation(I)
      lb <- m - alpha*sd/sqrt(n) /* Lower bound of the CI using formula 1 */
      ub <- m + alpha*sd/sqrt(n) /* Upper bound of the CI using formula 1 */
   Else /* type = sensor */
      m <- mean(M)
      sd <- standard_deviation(M)
      lb <- m - alpha*sd/sqrt(n*m) /* Lower bound of the CI using formula 1 */
      ub <- m + alpha*sd/sqrt(n*m) /* Upper bound of the CI using formula 1 */
    EndIf

    /* value(i) return current value of I or the average of each line of M depending of type */
    For i <- 1 to n do
      If (value(i) ∉ CI) Then
        Insert the current node i in SN EndIf
    EndFor
  Else
    If (n=3) /* TMR Redundancy */ Then
      If ((value(1) ≠ value(2)) || (value(2) ≠ value(3))) Then
        Central_Analysis /* because results of the 3 nodes are not equal */
      EndIf
    Else
      If (n=2) /* DMR Redundancy */ Then If (value(1) ≠ value(2)) Then Central_Analysis EndIf
      Else /* n=1: No redundancy */ Central_Analysis EndIf
    EndIf
  EndIf
End
```

Listing 1 Algorithm 1: Local_Analysis

updates the suspect nodes' confidence rates according to the result of the double homogeneity test. If the behaviour was erroneous, the central server then performs a correction using the generalized ESD test, which identifies the aberrant values in the data set in order to exclude them from future computations. The decision in Section 4.2.3 is thus based on corrected data.

These two processes, detection and correction, are detailed in Sections 4.2.1 and 4.2.2 respectively. To detect and potentially correct misbehaviour, the central server has to rely on data; therefore it has to build a set of data extracted from similar nodes installed at similar sites. This data set is an identification matrix (MI) for the identification nodes and a capture matrix (MC) for the sensing nodes. The central server has to request these data (along with the local CRs of the nodes if it does not have its own CR) from local servers at similar sites it has identified. To minimize the amount of data exchanged, the central server only requests data for the periods during which the nodes are suspected to have displayed failures and the local servers only provide data for the requested nodes (at most, data for the 3 nodes with the highest local CRs). A period is defined as an interval of time (for instance from 8h to 9h on a specific day).

For identification nodes:

$$MI = \begin{bmatrix} MI_{11} & \dots & MI_{1m} \\ \vdots & \ddots & \\ MI_{n1} & & MI_{nm} \end{bmatrix}$$

```
   Inputs : ident /* Identifier of suspect node or site */
            I /* Vector of values of the suspected node */
            alpha1 /* threshold of significance of the statistical test of double homogeneity */
            alpha2 /* threshold of significance of the ESD test */
 1 Begin
 2   Lss <- Search_Similar_Sites(ident) /* List of sites containing similar nodes to the suspect node */
 3   L <- High_Confidence_Rate(Lss) /* Use nodes with a high confidence rate */
 4   iRN <- Identification of node with the highest Confidence Rate in L /* id of Reference Node */
 5   m <- size(I) /* m is the number of periods */
 6   IR <- Create_Vector(m) /* Create a vector for value of the Reference Node */
 7   For i <- 1 to m do
 8     IR[j] <- Value returned by node iRN on period j
 9   EndFor
10   anomaly <- Detection (I,IR,alpha1) /* Boolean function */
11   If (anomaly = true) Then
12     n <- size(L) /* The number of nodes */
13     /* Compose M which can be MI or MC */
14     For i <- 1 to n do
15       For j <- 1 to m do
16         M[i][j] <- Value returned by node i on period j
17       EndFor
18     EndFor
19     Correction_Test_ESD(I,M,alpha2,m,A,AL)
20     /* Here output data (A, AL) can be used by central and/or local to exclude aberrant values of */
21     /* future computations: e.g. an accurate the decision process requires corrected data. */
22   EndIf
23   /* If required, here a central decision process can take place (see section 4.2.3) */
24 End
```

Listing 2 Algorithm 2: Central_Analysis

where MI_{ij} is the number of tagged assets read by the i^{th} identification node (having a high confidence rate) in the same period P_j.

For sensing nodes:

$$MC = \begin{bmatrix} MC_{11} & \dots & MC_{1m} \\ \vdots & \ddots & \\ MC_{n1} & & MC_{nm} \end{bmatrix}$$

where MC_{ij} is the mean of observations of the sensor S_i read by the different identification nodes (having read the sensor S_i and having a high confidence rate) in the same period P_j.

Each line of the matrix MC corresponds to the observations returned by a given sensor in different periods.

Algorithm 2 description Algorithm 2 is a pseudo-algorithm executed by the central server if a particular site requests validation of its data. The central server looks for all similar sites to the site containing the suspect nodes. For the analysis, the central server uses only sites with similar nodes that have high CRs. For generalization purposes, the matrix M in Algorithm 2 can be MI (identification matrix) or MC (sensing). It is built with values from similar nodes taken at the same periods. If an anomaly is detected in the behaviour of the suspect nodes by analyzing the matrix M with the Detection function, a correction can be done with the Correction_Test_ESD function using the ESD test which enables determination of aberrant values among the set of data, in order to exclude them from future computations. The details of the functions are given in the next two sub-sections (Listings 3 and 4).

As mentioned in line 23 of Algorithm 2, if decisions must be done at a central level, the matrix M (MI or MC depending of the type of nodes) with corrected data (i.e. without data detected as erroneous by the correction function) will be used in the decision process described in Section 4.2.3.

4.2.1 Detection

In order to detect whether there is a significant difference between the results of two similar nodes (a reference node and the suspect node) that are supposed to return similar results, we apply a statistical test for double homogeneity, known as Student's t-test, on two series of results of equal sizes and with unequal variances, since Student's t-test is highly robust for this case [25].

To determine whether there is a significant difference between the results from the nodes at a given threshold (1%, 5%, etc.), the central server thus calculates the random variable:

$$t = \frac{\overline{x_1} - \overline{x_2}}{\sqrt{\frac{1}{m}(\sigma_1^2 + \sigma_2^2)}} \tag{3}$$

where:

- $\overline{x_1}$ is the mean of the results of the reference node
- $\overline{x_2}$ is the mean of the results of the suspect node
- σ_1 is the standard deviation of the results of the reference node
- σ_2 is the standard deviation of the results of the suspect node
- m is the number of periods.

According to the conditions defined by the Student's-t test, if $m < 30$, the value μ_α is obtained from the Student-Fisher distribution for $k = 2m - 2$ degrees of freedom and at the threshold of significance α. If $m \geq 30$, the value μ_α is obtained from a table of normal distribution N(0,1) at the threshold of significance α.

Based on the comparison of $|t|$ and μ_α the server can decide:

- H_0, i.e. there is no significant difference between results from the two nodes (reference and suspect), if $|t| < \mu_\alpha$
- H_1, i.e. there is a significant difference between results from the two nodes (reference and suspect) if $|t| \geq \mu_\alpha$

Based on this result the CR at the central server is recomputed (in the case of H_1, it decreases, while in the case of H_0, it may increase). If H_1 applies, the fault detected on the suspect node must be corrected with the ESD test before any notification is made to the local server of the site. If H_0 applies, and the two nodes display similar behaviour, the local server of the site is notified of the suspect node's correctness.

The `Detection(I, IR, alpha)` function `Detection (I, IR, alpha)` is a function which has as inputs the vector `I` of the values of the suspect node, the vector `IR` of the reference node and the threshold of significance α (`alpha`) of the test. It returns a boolean to detect whether the suspect node is exhibiting erroneous behaviour or not by comparing its values with the values of the most reliable similar node (the reference node) in the same periods using the double homogeneity test.

4.2.2 Correction

In order to detect outliers among suspect nodes, we use the generalized Extreme Studentized Deviate [13, 14, 31] test. An outlying observation was defined by Grubbs [13] as one that appears to deviate markedly from other members of the sample in which it occurs. We have chosen the generalized ESD test since, compared to the Dixon test [9] or the Grubbs test [12], which can only detect one aberrant parameter, it can manage more aberrant parameters in a univariate data set (following approximately a normal distribution). The correction process thus consists of identifying erroneous data in order to remove these values from future computations.

```
Inputs :  I /* Vector of values of the suspect node */
          IR /* Vector of values of reference node */
          alpha /* threshold of significance */
Outputs : detection /* Boolean */
Begin
  m <- size(I) /* m is the number of periods */
  x1 <- mean(IR)
  x2 <- mean(I)
  sigma1 <- standard_deviation(IR)
  sigma2 <- standard_deviation(I)
  t <- (x1-x2)/sqrt((sigma1*sigma1+sigma2*sigma2)/m) /* Formula 3 */
  If (m<30) Then /* At the alpha significance level and at the 2*m-2 degrees of freedom */
    mu <- Student_Fisher_distribution(alpha,2*m-2)
  Else /* m>=30 */
    /* Value read in table of the normal distribution N(0,1) at the alpha significance level */
    mu <- Normal_distribution(0,1)
  EndIf
  If (absolute_value(t)<mu) Then detection <- false /* H0 */
  Else detection <- true /* H1 */ EndIf
End
```

Listing 3 Function: Detection(I, IR, alpha)

The `Correction_ESD_Test(I, M, alpha, r, A, AL)` function The `Correction_ESD_Test(I, M, alpha, r, A, AL)` has as inputs the vector `I` of the values of the suspect node, the matrix `M` of the values, the threshold of significance α (`alpha`) and the upper limit `r` of the aberrant values. The output provides the values suspected of being aberrant in the matrix for a vector `A` and a vector `AL` (of the same size as `A`), of which each element makes it possible to say whether the corresponding element of the vector `A` is an aberrant value or not.

First, the values of the matrix M and of I (the values of the suspect nodes) are put in a vector V. Then, `r` separate Grubbs' tests are performed on the set of values of V (there can be fewer than r tests according to the occurrence of several "extreme" values – more details are given in the algorithm of Listing 4):

$$G_i = \frac{max\{|x - \overline{x_i}|\}}{\sigma} \tag{4}$$

where:

- $x \in V$
- $\overline{x_i}$ is the average of the values of the vector V for the iteration i.
- σ is the standard deviation of V.

After having chosen G_i-$crit$ adjusted for the correct value of the sample size [31] (computed using the two-tailed inverse of the Student's-t distribution), if $G_i > G_i$-$crit$ then x, the value that maximize $|x - \overline{x_i}|$, is an outlier. If any test indicates that a value is aberrant then all the values declared non-aberrant previously are also outliers. Then all occurrences of the value x are removed from V and the whole process is repeated with the new V until `r` values (taking into account the number of occurrences of the removed values) have been removed from the original vector.

Simulation studies in [31] indicate that this critical value approximation, G_i-$crit$, is very accurate for $n \geq 25$ and reasonably accurate for $n \geq 15$ where n is the size of the original vector V.

4.2.3 Decisions

To enhance the accuracy of the decision-making process based on the collected and corrected data (as previously described), we propose the use of statistical hypothesis tests to minimize the risk of errors. A statistical hypothesis, known as confirmatory data analysis, is a hypothesis that is testable on the basis of observing a process that is modeled via a set

```
Inputs :  I /* Vector of values of the suspect node */
          M /* Matrix of identification or capture values */
          alpha /* threshold of significance */
          r /* The upper limit of aberrant values */
Outputs : A /* Vector of possible outliers */
          AL /* Vector of the logical values corresponding to the possible outliers */
Begin
  size <- nb_rows(M)*nb_columns(M)+size(I)
  V <- Create_Vector(size)
  /* Put I and M elements into vector V */
  k <- 1
  For i <- 1 to size(I) do
    V[k] <- I[i]
    k <- k+1
  EndFor
  For i <- 1 to nb_rows(M) do
    For j <- 1 to nb_columns(M) do
      V[k] <- M[i][j]
      k <- k+1
    EndFor
  EndFor
  A <- Create_Vector(r)
  AL <- Create_Vector(r)
  For i <- 1 to r do
    min <- min(V) /* The minimum value of the vector V */
    max <- max(V) /* The maximum value of the vector V */
    m <- mean(V) /* The mean of values of the vector V */
    stdv<- standard_deviation(V) /* The standard deviation of values of the vector V */
    min_mean <- m - min
    max_mean <- max - m
    G <- max(min-mean,max-mean)/stdv
    sig_value <- alpha/size
    df <- size - 2
          /* tinv is two-tailed inverse of the student's t-distribution */
          /* for (Probability, Degrees of freedom) */
    t_crit <- tinv(sig_value,df)
    G_crit <- (size-1)*t_crit/sqrt(size*(df+t_crit*t_crit))
    If (min_mean>max_mean) Then
      A[i] <- min
    Else
      A[i] <- max
    EndIf
    If (G>G_crit) Then
      AL[i] <- true
      For j <- 1 to i-1 do AL[i] <- true /* All previous values are outliers */ EndFor
    Else AL[i] <- false
    EndIf
    nbo <- occurrences of A[i]
    size <- size - nbo
    Remove all occurrence of A[i] from V /* Eliminate the possible aberrant value from vector V */
    i <- i + nbo
  EndFor
End
```

Listing 4 Function: Correction_ ESD_Test(I, M, alpha, r, A, AL)

of random variables [36]. This analysis, which can be applied both at a local level and at the central level, will help to improve the dependability of C-IoT by providing a more reliable way to make decisions (such as raising an alarm because an event is detected, or deciding to replace faulty nodes). To minimize the error rate, rather than basing the decision process on only a single comparison with a threshold, with statistical hypothesis testing decisions are based on computation of a Z-score to choose more reliably between the null hypothesis (H_0) and the alternative hypothesis (H_1). The null hypothesis can be defined as the case where there is nothing to do (the data are correct). The alternative hypothesis is what we aimed to test. We illustrate this process using two cases. The first consists of applying the statistical hypothesis test [29] to a matrix of collected data from homogeneous sensing nodes to determine the occurrence of an event based on the average. In the second case, the statistical hypothesis test is applied to the CR [26] of a node to decide about its state (defective or not).

Case 1: Decision process based on the average of collected data to determine the occurrence of an event Using a matrix M of size $n \times m$ of data collected by n sensing nodes during m periods, if $n * m \geq 30$, the distribution is normal, and the average μ_{H_0} of the observed phenomena is known, we can formulate a hypothesis to check at a given significance level α (1%, 5%, etc.): H_0: $\overline{x} \leq \mu_{H_0}$ and H_1: $\overline{x} > \mu_{H_0}$ where $\overline{x}$ is the mean of M.

For instance, for temperature data, the average may be $\mu_{H_0} = 20°C$ and H_0 would be that the sensed temperature data are compliant with normal behaviour, whereas H_1 would mean the occurrence of an event (a significantly higher temperature is detected). The Z-score is then computed with:

$$Z_{calculated} = \frac{\overline{x} - \mu_{H_0}}{\frac{\sigma}{\sqrt{n*m}}} \tag{5}$$

where σ is the standard deviation of M. The chosen α, Z_α is obtained from a table of normal distribution N(0,1) at α.

Then, H_0 is maintained if $Z_{calculated} \leq -Z_\alpha$ and H_0 is rejected (and thus H_1 is chosen) if $Z_{calculated} > -Z_\alpha$. In the case of H_0, the system has nothing to do and in the case of H_1, the system can trigger appropriate action to handle the event.

Similar test Z-scores can be computed using metrics other than the average.

Case 2: Decision process using the confidence rate of a node to determine its state In a similar manner, instead of deciding that a node is defective when its CR is below a defined threshold π_0, the Z-score can be computed at a significance level α (1%, 5%, etc.) using the following formula:

$$Z_{calculated} = \frac{CR - \pi_0}{\sqrt{\frac{\pi_0(1-\pi_0)}{n}}} \tag{6}$$

Table 1 Comparison of different approaches for IoT and C-IoT according to data analysis, data correction and decision making

Work	Domain	Data Analysis	Data Correction	Decision
Boano et al. [5, 30]	IoT	No	No	No
Objective: increase reliability of IoT solutions in the presence of environmental interferences				
Techniques: design testbeds to examine the impact of environmental interferences				
Macedo et al. [24]	IoT	No	No	No
Objective: estimate reliability and availability of IoT applications by considering redundancy aspects				
Techniques: mathematical models based on Markov chains				
Dar et al. [8]	IoT	No	No	No
Objective: build a framework based on the concept of virtualized IoT services				
Techniques: dependability by design				
Borges et al. [6]	C-IoT	Yes	Yes	Partial
Objective: select data from reliable sensors				
Techniques: collaborative filtering techniques to detect; refinement of selected data to correct; decision consists of selecting the most reliable sensor from a set				
Cons: address only sensors				
Our proposal	C-IoT	Yes	Yes	Yes
Objective: enhance dependability with profiling techniques				
Techniques: Confidence Interval, Confidence Rate and double homogeneity test to detect misbehaviour and Extreme Studentized Deviate (ESD) test to detect outliers; outliers detected by ESD test can be excluded from future computations to support correction; decision processes are supported by statistical hypothesis tests as illustrated in Section 4.2.3				
Pro: address identification and sensing nodes based on different technologies				

where CR is the confidence rate of the node and n is the total number of tests used to compute CR in formula (2). In this case, H_0: $CR \geq \pi_0$ (i.e. the node is functioning properly) and H_1: $CR < \pi_0$ (i.e. the node is defective). The chosen α, Z_α are obtained from a table of normal distribution N(0,1) at α.

H_0 is maintained if $Z_{calculated} \geq -Z_\alpha$ and H_0 is rejected (and thus H_1 is chosen) if $Z_{calculated} < -Z_\alpha$. In the case of H_0, the system has nothing to do and in the case of H_1, the system can decide to notify an operator to check the node, reconfigure it, repair it, or even replace it.

Although errors in decision-making cannot be completely eliminated, we can minimize them by using corrected data and the appropriate level of significance.

4.2.4 Comparison with related work

To highlight our contributions, Table 1 summarizes the approaches proposed in work related to IoT and C-IoT mentioned respectively in Sections 2.2 and 2.3 with regard to the different steps identified: i.e. data analysis, data correction and decision making.

5 Conclusion

In this paper, we have presented an approach based on several profiling processes using statistical data analysis to enhance the dependability of the future C-IoT. First, we defined the C-IoT and provided an architectural blueprint involving several distributed sites collaborating to improve dependability, using well-known technologies of the IoT (RFID, NFC and beacons). Then, using the double homogeneity test, we showed how to detect potential misbehaviour by profiling the suspect node against a reference. Using the generalized ESD test, by comparing data from a suspect node with data from homogeneous nodes, we identified outliers to exclude from future computations to support a more accurate process of decision making. In addition, we used a third statistical data analysis method, the statistical hypothesis test, to minimize decision-making risks. The strength of our proposal is that it can be extended to any types of devices. Altogether, these simple statistical data analysis approaches have the potential to create more reliable complex systems such as the C-IoT or any multimedia-aware IoT systems. In addition, the exponential increase in connected devices, with more embedded sensors, will support our approach by making analysis more efficient and accurate.

Our future work will involve developing and deploying a testbed for such a C-IoT architecture with additional technologies in order to test new statistical data analysis methods under operational conditions and to fine-tune the values of parameters such as levels of significance, threshold, and `r`, used in the different tests.

References

1. Beacon A ASensor. http://wiki.aprbrother.com/wiki/ASensor. Visited on 2017-10-20
2. Beacon A April Beacon website. https://blog.aprbrother.com/. Visited on 2017-10-20
3. Behmann F, Wu K (2015) Collaborative internet of things (C-IoT): for future smart connected life and business. Wiley
4. Belkacem I, Bahloul SN, Aktouf OEK (2014) Data analysis of an RFID system for its dependability. Int J Embedded Real-Time Commun Syst (IJERTCS) 5(3):1–22

5. Boano CA, Römer K, Voigt T (2015) RELYonIT: dependability for the internet of things. IEEE IoT Newsl 13
6. Borges Neto JB, Silva TH, Assunĉão RM, Mini RA, Loureiro AA (2015) Sensing in the collaborative internet of things. Sensors 15(3):6607–6632
7. Chavira G, Nava SW, Hervas R, Bravo J, Sanchez C (2007) Combining RFID and NFC technologies in an AmI conference scenario. In: Eighth Mexican International conference on current trends in computer science, 2007. ENC 2007. IEEE, pp 165–172
8. Dar KS, Taherkordi A, Eliassen F (2016) Enhancing dependability of cloud-based IoT services through virtualization. In: 2016 IEEE First international conference on internet-of-things design and implementation (IoTDI). IEEE, pp 106–116
9. Dean RB, Dixon W (1951) Simplified statistics for small numbers of observations. Anal Chem 23(4):636–638
10. Fritz G, Beroulle V, Nguyen M, Aktouf OEK, Parissis I (2010) Read-error-rate evaluation for RFID system on-line testing. In: 2010 IEEE 16th international mixed-signals, sensors and systems test workshop (IMS3TW). IEEE, pp 1–6
11. Fritz G, Beroulle V, Aktouf OEK, Hely D Méthodes statistiques pour le test en ligne des systèmes rfid uhf https://www.researchgate.net/profile/Oum-El-Kheir_Aktouf/publication/268304220_Methodes_statistiques_pour_le_test_en_ligne_des_systemes_RFID_UHF/links/54be8aff0cf28ce312326cfe/Methodes-statistiques-pour-le-test-en-ligne-des-systemes-RFID-UHF.pdf. Visited on 2017-10-20
12. Grubbs FE (1950) Sample criteria for testing outlying observations. Ann Math Stat 27–58
13. Grubbs FE (1969) Procedures for detecting outlying observations in samples. Technometrics 11(1):1–21
14. Grubbs FE, Beck G (1972) Extension of sample sizes and percentage points for significance tests of outlying observations. Technometrics 14(4):847–854
15. Hamdan D (2013) Détection et diagnostic des fautes dans des systèmes à base de réseaux de capteurs sans fils. Ph.D. thesis, Université de Grenoble
16. Herrera MM, Bonastre A, Capella JV (2008) Performance study of non-beaconed and beacon-enabled modes in IEEE 802.15.4 under bluetooth interference. In: The Second international conference on mobile ubiquitous computing, systems, services and technologies, 2008. UBICOMM'08. IEEE, pp 144–149
17. Intel A guide to the internet of things: how billions of online objects are making the world Wise. https://www.intel.com/content/www/us/en/internet-of-things/infographics/guide-to-iot.html. Visited on 2017-10-20
18. IV P PHASE IV website. https://www.phaseivengr.com. Visited on 2017-10-20
19. Kalia M, Garg S, Shorey R (2000) Efficient policies for increasing capacity in Bluetooth: an indoor pico-cellular wireless system. In: Vehicular technology conference proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st, vol 2. IEEE, pp 907–911
20. Kajioka S, Mori T, Uchiya T, Takumi I, Matsuo H (2014) Experiment of indoor position presumption based on RSSI of Bluetooth LE beacon. In: 2014 IEEE 3rd Global conference on consumer electronics (GCCE). IEEE, pp 337–339
21. Kendall MG, Stuart A (1969) The advanced theory of statistics - v2 Inference and relationship. Griffin, London
22. Kevin A (2009) That 'Internet of Things' thing, in the real world things matter more than ideas. RFID J 22
23. Laprie JC, Arlat J, Blanquart J, Costes A, Crouzet Y, Deswarte Y, Fabre J, Guillermain H, Kaâniche M., Kanoun K et al (1995) Guide de la sûreté de fonctionnement. Cépaduès, Toulouse
24. Macedo D, Guedes LA, Silva I (2014) A dependability evaluation for internet of things incorporating redundancy aspects. In: 2014 IEEE 11th International conference on networking, sensing and control (ICNSC). IEEE, pp 417–422
25. Markowski CA, Markowski EP (1990) Conditions for the effectiveness of a preliminary test of variance. Am Stat 44(4):322–326
26. Merrill RM (2012) Fundamentals of epidemiology and biostatistics. Jones & Bartlett Publishers
27. Mtita C (2016) Lightweight serverless protocols for the internet of things. Institut National des Télécommunications, Ph.D. thesis
28. NXP Freescale home health hub reference platform. https://www.nxp.com/docs/en/fact-sheet/HMHLTHHUBFS.pdf. Visited on 2017-10-20
29. Paulson DS (2003) Applied statistical designs for the researcher. CRC Press
30. RELYonIT Research by experimentation for dependability on the internet of things. http://www.relyonit.eu. Visited on 2017-10-20
31. Rosner B (1983) Percentage points for a generalized ESD many-outlier procedure. Technometrics 25(2):165–172

32. Sample A, Zhao Y NFC-WISP website. https://nfc-wisp.wikispaces.com/. Visited on 2017-10-20
33. Schaffers H, Komninos N, Pallot M, Trousse B, Nilsson M, Oliveira A (2011) Smart cities and the future internet: towards cooperation frameworks for open innovation. Fut Int 431–446
34. Semiconductor N nRF51822 Bluetooth Smart Beacon Kit. http://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF51822-Bluetooth-Smart-Beacon-Kit. Visited on 2017-10-20
35. Semiconductor N Nordic Semiconductor website. http://www.nordicsemi.com/. Visited on 2017-10-20
36. Stuart A, Ord JK, Arnold S (1999) Kendall's advanced theory of statistics. Vol 2A: classical inference and the linear model, vol 2. Edward Arnold, London
37. Thinfilm Thinfilm website. http://www.thinfilm.no/. Visited on 2017-10-20
38. Thornton F, Sanghera P (2011) How to cheat at deploying and securing RFID Syngress
39. WISP Home https://wisp5.wikispaces.com/WISP+Home. Visited on 2017-10-20
40. Zhao Y, Smith JR, Sample A (2015) Nfc-wisp: a sensing and computationally enhanced near-field rfid platform. In: 2015 IEEE International conference on RFID (RFID), pp 174–181. https://doi.org/10.1109/RFID.2015.7113089

Imad Belkacem Imad is a PhD student working as a teacher at the University of Mostaganem (Algeria). Prior to beginning the PhD program, he undertook core courses in computer science at the National Institute of Informatics. He completed his diploma (Engineer in Computer Science) at the University of Mostaganem and his Magister diploma at the University of Oran (Algeria). During his studies, he developed an interest in how Service Oriented Architecture (SOA) promises a major change in the design of company information systems. He is currently working on a research project in the LITIO Laboratory at the University of Oran that explores how Fault Tolerance increases the dependability of an RFID (Radio Frequency Identification) system.

Safia Nait Bahloul obtained her doctorate in Computer Science from the University of Oran. Since 2011, she has been a member of the LITIO laboratory at the University of Oran, which was accredited in 2009. She manages a research team in the LITIO laboratory on data engineering and Web technology. Since 2008, she has also been responsible for an academic master's degree in Information Systems and Web Technology. Her research focuses on advanced aspects of databases, web technology and unsupervised classification. Her work has been published in several journals and conference proceedings. She has supervised several doctoral and masterate candidates and undergraduate projects in the field of information research, Clustering, MDA and security (access control).

Damien Sauveron received his MSc and PhD degrees in Computer Science from the University of Bordeaux, France. He has been Associate Professor with Habilitation at the XLIM Laboratory (UMR CNRS 7252, University of Limoges, France) since 2006. He is Head of the Computer Science Department in the Faculty of Science and Technology at the University of Limoges. Since 2011, he has been a member of the CNU 27, the National Council of Universities (for France). He has been chair of IFIP WG 11.2 Pervasive Systems Security since 2014, having previously been appointed vice-chair of the working group. His research interests are related to smart card applications and security (at hardware and software level), RFID/NFC applications and security, mobile network applications and security (particularly UAV), sensor network applications and security, Internet of Things (IoT) security, cyber-physical systems security, and security certification processes. In December 2013, the General Assembly of IFIP (International Federation for Information Processing) awarded Dr Sauveron the IFIP Silver Core award for his work. He has been involved in more than 100 research events in a range of capacities (including PC chair, General Chair, Publicity Chair, Editor/Guest Editor, Steering Committee member, and Program Committee member).