Journal of
Sensor and Actuator Networks

MDPI

Article

Design and Implementation of a CoAP-Compliant Solution for RFID Inclusion in the Internet of Things

Ivan Farris, Sara Pizzi, Antonella Molinaro and Antonio Iera *

DIIES, University "Mediterranea" of Reggio Calabria, Via Graziella, Loc. Feo di Vito, 89100 Reggio Calabria, Italy; ivan.farris@unirc.it (I.F.); sara.pizzi@unirc.it (S.P.); antonella.molinaro@unirc.it (A.M.)
* Correspondence: antonio.iera@unirc.it; Tel.: +39-0965-875-3247

Academic Editor: Michele Magno
Received: 25 June 2016; Accepted: 19 October 2016; Published: 1 November 2016

Abstract: Recent technological advancements allowed widening the applicability scope of the RFID (Radio Frequency Identification) technology from item identification to sensor-enabled computation platforms. This feature, added to the native radio energy-harvesting capability and the extremely low power consumption, has attracted the interest of research and industrial communities and pushed them to include the RFID technology into a global network of interconnected objects, as envisaged by the Internet of Things paradigm. In the last few years, standardization bodies have made significant efforts to design lightweight approaches, such as CoAP (Constrained Application Protocol), to efficiently manage resource-constrained nodes by using traditional web interfaces; nevertheless, RFID integration is not addressed yet. In this paper, we propose a CoAP-compliant solution where RFID tags, behaving as virtual CoAP servers, are directly accessible by remote CoAP clients via a reader, which acts as a CoAP proxy. A real testbed, addressing key aspects, such as tag addressing, discovery and management of CoAP requests via RFID operations, is deployed to validate the feasibility of the proposal. Experimental results show rapid response times: less than 60 ms are requested for resource retrieval, while from 80 to 360 ms for sending data to the RFID device, depending on the tag memory dimension.

Keywords: Internet of Things; RFID technology; wireless sensor networks; Constrained Application Protocol

1. Introduction

Radio Frequency Identification (RFID) technology has experienced a continuous evolution in the last decade, by moving from simple transponders for tracking purposes to sensor-equipped smart tags, which are able to implement sensing and processing functions [1–3]. In the broad landscape of wireless sensor networks, RFID-based devices present unique features, which make them extremely attractive. First of all, their completely passive radio interface allows for zeroing power consumption during data exchanges (when wireless sensor devices usually dissipate a large amount of energy). Besides, the RFID native energy harvesting feature can be exploited to charge a fully-programmable Micro Control Unit (MCU) and some embedded sensors. As a consequence, many studies confirm that battery-less RFID tags can be very helpful in indoor environments to support smart home [4] and e-health [5] applications. Moreover, equipping the RFID sensor tag with a capacitor for long-run sensing activities [6,7] paves the way to a wider range of applications, such as data logging or environmental sensing also in wide-range scenarios [8,9].

However, to fully exploit the potential offered by RFID-based devices, their integration into the Internet of Things (IoT) landscape needs to be carefully addressed. The main objective of IoT is to implement a world-wide network wherein uniquely addressable heterogeneous devices can exchange information and share services [10].

Enabling the extension of Internet technologies to resource-constrained devices will allow for the integration of several, and sometimes far apart, technological worlds. In this view, a fervent research activity has been carried out by some Working Groups (WGs) of the Internet Engineering Task Force (IETF), such as 6LoWPAN, 6lo, Routing Over Low power and Lossy networks (ROLL), and Constrained RESTful Environments (CoRE), to define IoT standards. These groups, supported in great part of the IoT research community, aim at exploiting the vast pre-existing IP-related experience to implement a global network of constrained nodes, leveraging two principles: (i) all of the things are IPv6 addressable; and (ii) services offered by devices can be accessed in a RESTful manner. Specifically, activities of the Constrained RESTful Environments (CoRE) group are currently focusing on the standardization of an application protocol for the manipulation of resources on a device, the Constrained Application Protocol (CoAP) [11].

Despite the efforts to develop a globally-recognized standard for the transparent interoperability of intranets of objects, the integration of RFID technology in the IoT domain still requires further attention. Actually, a multitude of deployed tags still constitute isolated intranets since either they do not have enough hardware capabilities to support the proposed approach, or they use a proprietary protocol to exchange data with the reader. For these groups of RFID devices, which we refer to as "legacy" tags, no solutions for seamless integration in IoT has been designed yet. Indeed, the cited integration will happen only if the designed procedures will enable "real-time" direct communications with RFID tags and same access to their resources as to resources of any other IoT object (such as wireless sensors, for example).

In this paper, we contribute to the mentioned objective by providing a CoAP-compliant solution where RFID tags are managed as a sort of "virtual CoAP server" that allows their own resources to be directly accessible from CoAP clients through the reader. To this aim, we foresee a CoAP protocol implementation on augmented RFID readers, referred to as "CoAP-RFID proxy readers", that act as proxies for the RFID tags, by intercepting requests from the CoAP clients and answering on behalf of the tags.

In a preliminary work [12], we addressed the key design aspects of "CoAP-RFID proxy readers" and we provide a preliminary analysis. In that work, we addressed several challenges, such as: (i) tag addressing, through the mapping of tag identifiers into IPv6 addresses; (ii) service and resource discovery to enable the dynamic search for RFID resources in IoT; (iii) mapping of CoAP methods onto RFID commands.

In this paper, we extend the cited research by providing the following contributions:

- discuss the most promising integration approaches of the RFID technology in the IoT domain and evaluate the pros and cons of each solution;
- analyze in detail how the "CoAP-RFID proxy reader" works in a personal smart health scenario, by describing the interaction among heterogeneous devices in such an environment;
- extend Californium [13], an open-source Java CoAP framework, to implement a prototype of the "CoAP-RFID proxy reader", by addressing the management of the tags as "virtual CoAP servers" and relevant RFID resources;
- present the results obtained from measurements in a real testbed deployed to assess the performance of the implemented solution.

The paper is organized as follows. In Section 2, we provide a comprehensive overview of relevant works in the areas of the RFID technology and the Web of Things paradigm. Section 3 describes the most promising integration approaches and contextualizes our proposal in the research literature, by underling envisaged benefits. The designed solution is described in Section 4. Section 5 sets out in detail how our proposal works in a sample application scenario. Finally, Section 6 describes the implementation of the "CoAP-RFID proxy reader", whereas the testbed setup and the experimental results are discussed in Section 7. Section 8 concludes the paper.

2. Related Works and Research Background

This section addresses the state-of-the-art of the research activities conducted in the areas of interest for our research. We first focus on the RFID technology to provide detailed information about communication protocols standardized for both UHF readers and tags. Then, we thoroughly browse relevant literature works that introduce several schemes to enable resource retrieval of sensor-equipped RFID tags, in the absence of a standardized solution. Last, we introduce the solutions to access resources of constrained devices by adopting the so-called Web of Things paradigm. Attention is paid to the description of the CoAP protocol and to the core concepts to enable web interfaces for IoT nodes.

2.1. RFID Internetworking

To boost a world-wide diffusion of the RFID technology, EPCglobal and the International Organization for Standardization (ISO) have published standards addressing aspects of radio communication, application interface, data encoding, network design, etc.

According to EPCglobal, tags with different complexity levels and functionalities can be read by RFID readers: Class 1 refers to identity-tags, which store only an identifying code; Class 2 includes tags with additional memory storage and, optionally, with sensing capability, such as Wireless Identification and Sensing Platforms (WISPs) [1]; Class 3 specifies battery-assisted tags that communicate passively and use an on-board power source to energize sensors; and Class 4 tags exploit the battery also to communicate.

The radio communication between RFID readers and tags is regulated by the Air Interface standards, EPC Gen-2 protocol [14] and ISO 18000-63 [15]. These standards are aligned on the core functionalities and define modulations, encoding, medium access schemes and a set of basic commands. Three categories of commands are foreseen: Select, allowing the reader to choose the subset of tags to handle; Inventory, allowing the reader to acquire the EPC codes stored in their memory; Access, allowing the reader to perform operations, such as reading and writing data, locking memory sections and even disabling tags.

On the other hand, application protocols define the message structure and the modalities for interacting with RFID readers. To overcome interoperability problems and to facilitate the development of applications, communications can happen at different abstraction levels. The Application Level Events (ALE) protocol [16] specifies a software interface through which client applications may interact with filtered and consolidated EPC data. Typically an RFID middleware implements the ALE interface, but also smart RFID readers can provide it. Using ALE, an application generates a high-level description of, for example, the data to read from or to write to tags, the timing of operations and the filters to select particular tags. Alternatively, the Low Level Reader Protocol (LLRP) [17] provides specific parameters and controls to set the command and timing parameters of the RFID air protocol. The LLRP reader protocol does not allow for simultaneous requests from multiple devices, and this reduces its performance in pervasive environments. Moreover, it does not foresee any specific feature to efficiently support sensor-equipped tag communication.

A few works have addressed the integration of the RFID Technology into IoT at the network level so far. In [18], a common addressing scheme is proposed that extends IPv6 to the RFID tags. Some IPv6-based initiatives, such as the IoT6 project [19], have attempted to include the EPCglobal platform [20] into their architecture; the objective of the integration is only the access to information stored in EPCglobal distributed databases. For next-generation computational RFID tags, in [21], we also proposed a novel communication paradigm that enables native IPv6 internetworking, by implementing an IPv6-compliant protocol stack in the enhanced tags. On the other hand, the IETF is driving the standardization of IPv6-based protocols for the IoT. In particular, the 6lo working group (WG) [22], starting from the 6LoWPAN specifications, has been created to enable IPv6 connectivity for a variety of link-layer technologies, such as Bluetooth [23], DECT (Digital Enhanced Cordless Telecommunication) [24] and NFC [25]. However, the 6lo standardization work has not

yet comprehensively considered RFID among the analyzed link-layer technologies, since solutions designed for NFC are unfortunately unsuited to UHF Gen-2 RFID systems due to differences in the HF radio interface, the need for powering devices in NFC peer-to-peer mode and native unicast NFC transmission.

As for the "sensing capability" of RFID devices, the literature is rich in solutions proposed to transfer sensing data generated by RFID tags. In [1], the sensor information is encoded in a portion of 96-bit EPC, thereby allowing data exchanges during inventory. Another approach implemented by the majority of smart RFID tags exploits the user memory to store the samples generated by embedded sensors, so that the reader can later retrieve them by issuing a Read Command. In [2], an organization of the sensor readings in 72-bit frames is proposed to enable the sequential delivery of measurement reports. A reconfigurable UHF RFID sensing tag is designed in [3], which allows one to dynamically manage the integrated sensors, by defining a specific scheme for the tag user memory. The reader can modify the sampling period or the operating mode of sensors, by updating the memory fields. The dynamic management of RFID tags could extend their applicability scope to fields typically covered by wireless sensor networks, enabling efficient sensing and monitoring activities [26–28].

In [29], smart RFID tags could exchange data inside the coverage area of an RFID reader, by suitably managing the user memory and defining specific procedures to collect and forward messages. The concept of tag memory as a virtual communication channel has also been studied in [30,31], where the available memory of surrounding tags is used to enable communication among a group of readers.

All of the solutions for sensor-based RFID tags mentioned above refer to specific tag platforms and do not guarantee the interoperability needed to enable a seamless deployment of RFID systems in the IoT domain. Therefore, a major challenge is still represented by the design of methods to make the RFID tag resources accessible by heterogeneous remote clients in a standardized way.

2.2. Connecting Objects to the Web through CoAP

One step beyond the network connectivity foreseen by the IoT vision, the Web of Things initiative aims at achieving the integration of "smart things" not only into the Internet (i.e., at the network layer), but also into the web (i.e., at the application layer). The Web of Things can be realized through the Representation State Transfer (RESTFul) architectural style [32], a powerful mechanism to build communication interfaces and protocols over the World Wide Web that enable information exchange and interoperability among systems.

CoAP is a Web transfer protocol that follows the REST architectural style, currently under development by the IETF CoRE WG for use with constrained nodes and low power lossy networks [11]. It provides a request/response interaction model between application endpoints and supports built-in discovery of services and resources. A CoAP endpoint acts as a client when it originates a request or as a server when it answers a request.

The resource discovery offered by a CoAP endpoint is a distributed mechanism where a CoAP client queries a server for its list of hosted resources. CoAP resources are identified by Universal Resource Identifiers (URIs) and can be acted upon by CoAP methods. Similarly to HTTP, four methods are specified for a CoAP request: GET (retrieves the resource), POST (processes a resource), PUT (updates/creates a resource) and DELETE (deletes a resource). Deciding which resources are made discoverable (if any) is up to the CoAP server. Clients can also query for specific types of CoAP resources. This is achieved by utilizing a query string in the request method consisting of search parameters listed as parameter=value pairs. Once the list of available resources is obtained from the server, the client can send further requests to retrieve the value of a certain resource.

In many IoT scenarios, direct discovery of resources is inefficient due to the presence of sleeping nodes, disperse networks or networks where multicast traffic is ineffective. These problems can be solved by employing an entity called a Resource Directory (RD) [33], which hosts descriptions of

resources held on other servers and implements a REST interface that allows registration and lookup of those resources. Endpoints are assumed to proactively register and maintain resource directory entries on the RD, which are soft state and need to be periodically refreshed. It is also possible for an RD to proactively discover resources from endpoints and add them as resource directory entries or to validate existing resource directory entries. A lookup interface for discovering any of the resources held in the RD is provided using the CoRE Link Format [34].

3. Approaches for RFID Inclusion in the IoT

Figure 1 schematically depicts the three approaches for RFID inclusion in the IoT that will be discussed in the remainder of this paper.

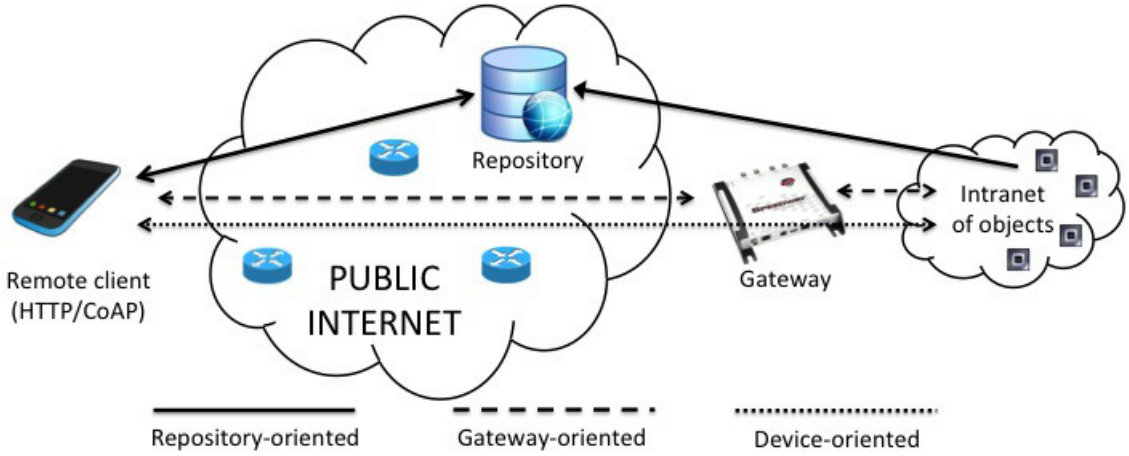


Figure 1. Representation of the three discussed approaches for RFID inclusion in the IoT.

3.1. Repository-Oriented

The first architecture envisages the use of repositories to store the information relevant to legacy RFID tags, i.e., "traditional" low-cost tags only equipped with user memory. When a user is interested in retrieving information about a device, he/she issues a request to the appropriate repository, which typically exposes the devices resources through web services. The main advantages of this solution are: (i) the possibility to filter and aggregate the big amount of raw data produced by devices and to store only meaningful events; (ii) reusing the existing database technologies; (iii) decoupling the application logic from the issues of collecting data; and (iv) maintaining historical data and events for long-term processing. However, this solution does not allow the user to establish an effective real-time communication with the device, and it is really difficult to choose the granularity of data to store if the use case is not a priori defined. An excessive level of details can make the size of the database explode, while excessive filtering and aggregation might imply the loss of precious information.

The repository-oriented approach follows the philosophy of the actual EPCglobal architecture [20], which has been designed to support the global supply chain and is considered as the infancy of IoT. Accordingly, business events relevant to RFID-tagged objects are stored in the EPCIS (Electronic Product Code Information Service) repositories, and the product lifecycle can be tracked from the manufacturer to the final customer. However, albeit the efforts undertaken so far (e.g., in [35,36]) to extend the EPCglobal architecture to devices other than RFID tags and readers, the integration process has been slowed down by the task complexity and the standardization procedures.

3.2. Gateway-Oriented

A different approach, named "gateway-oriented", foresees that the user explicitly queries a server that offers a predefined set of APIs (Application Programming Interfaces) to access the resources of the legacy devices. The gateway is typically located at the edge of the network and translates standard Internet protocols into proprietary protocols. This approach enables real-time interaction with the device, although the client needs to explicitly address its requests towards a third entity like the gateway. A further benefit is the opportunity to demand of the gateway efficient query processing, such as generating average, maximum and minimum values of the data produced by devices.

In the RFID domain, the gateway-oriented approach translates into the interaction with the reader to retrieve information about tags within its coverage range. Various protocols have been standardized, at different levels of abstraction, such as ALE or LLRP. They have been mainly designed for tracking purposes and offer advanced features to report tag inventory, but should be updated to support the novel sensor-enabled RFID tags.

3.3. Device-Oriented

Another approach is to directly communicate with the device. Thus, it is referred to as "device-oriented". This solution is under investigation by the IETF working groups that promote an IoT lightweight protocol stack to end-to-end interaction with resource-constrained devices, but not with RFID tags.

We think that the device-oriented approach for RFID devices can be effectively implemented by two different architectures. The first option is that devices natively support CoAP and can directly reply to the client requests. Thus, the border router, which connects the RFID network to the Internet, acts as an enhanced edge IPv6 router by forwarding and proactively retrieving messages directed to tags. Optionally, it could perform simple protocol translation operations (such as HTTP-CoAP), but it is not required to understand the query or interpret the content of the request. In [21], we designed a specific adaptation layer to natively support IPv6 communication for RFID smart tags with storage and computation capabilities. It represents a first step towards the full integration of CoAP-compliant RFID system in the Web of Things.

Although this solution is highly desirable in the near future, it requires smart tags, while almost the totality of current standard RFID tags can only execute simple reading and writing commands issued by the RFID readers. Furthermore, current sensor-equipped tags use proprietary memory management schemes to exchange sensing information, which severely limit the full IPv6 interoperability in heterogeneous environments. Therefore, a more convenient way to integrate legacy RFID devices following an alternative device-oriented approach is to develop an architecture where a reader acts as a transparent proxy and provides a CoAP interface bound to a virtual IPv6 address for each tag. In doing so: (i) a remote CoAP client can look for the resources exposed by any type of object and establish a direct communication with both native CoAP devices and legacy devices residing behind a proxy; and (ii) legacy devices can interact with each other through standardized IoT interfaces. This solution represents a novel hybrid approach to include resource-constrained devices in the IoT landscape. Even if the reader implements the integration logic, the identities of the single devices, i.e., the tags, are preserved by their association to the corresponding virtual CoAP endpoints. Thus, the end-to-end principle, which is the basis of the global IPv6 internetworking, is respected. In this paper, we investigate the device-oriented approach, by developing a transparent "CoAP-RFID proxy Reader" able to manage the tags as "virtual CoAP servers" and to access the relevant tags' resources.

Table 1 shows a comparison among the discussed feasible integration approaches by referring to the main requirements for an efficient integration of the RFID technology in the IoT.

Table 1. Comparison among the discussed integration approaches. RD, Resource Directory.

Requirement	Repository-Oriented	Gateway-Oriented	Device-Oriented	
			Native	Transparent Proxy
Historical data	X			
Real-time interaction		X	X	X
Tag inventory		X	with RD	with RD
Read/Write Operation		X	X	X
End-to-End Communication			X	X
Support of legacy devices				X

4. The Proposed Architecture for RFID Inclusion in the IoT

In the IETF terminology, legacy RFID tags could be considered Class 0 devices [37], since they can participate in Internet communications only with the help of more powerful devices (i.e., RFID readers), acting as proxies or gateways, and are unable to host a protocol stack specifically designed for constrained nodes (e.g., CoAP over UDP).

In order to integrate the RFID technology in the IoT scenario, we thus propose to extend the RFID reader functionality with CoAP proxy capability in order to manage the interaction with tags. In the conceived architecture, each RFID tag is considered as a sort of virtual CoAP server that advertises its own resources via the reader. Tags do not terminate CoAP traffic directly, but communicate with the reader, which transparently represents the tag (physical object) for a remote client, since it intercepts all of the requests sent to the URIs of tags and replies on their behalf. In this way, we adhere to a device-oriented integration philosophy, as described in the previous section. The reader also acts as a Resource Directory (RD) [33] for the managed tags; it stores the description of tags' resources and allows remote lookup for these resources. Thus, tags' inventory can be performed by using a CoAP logic, i.e., by simply requesting of the RD the addresses of all of the nodes in a specific domain.

Emulating web service interfaces between CoAP endpoints guarantees the respect of the basic principle of end-to-end connectivity, thus enabling communications between constrained nodes with different technologies. The use of specific HTTP-CoAP proxies [38] enables access to tag resources also from HTTP clients and fosters the deployment of web applications (see Figure 2).

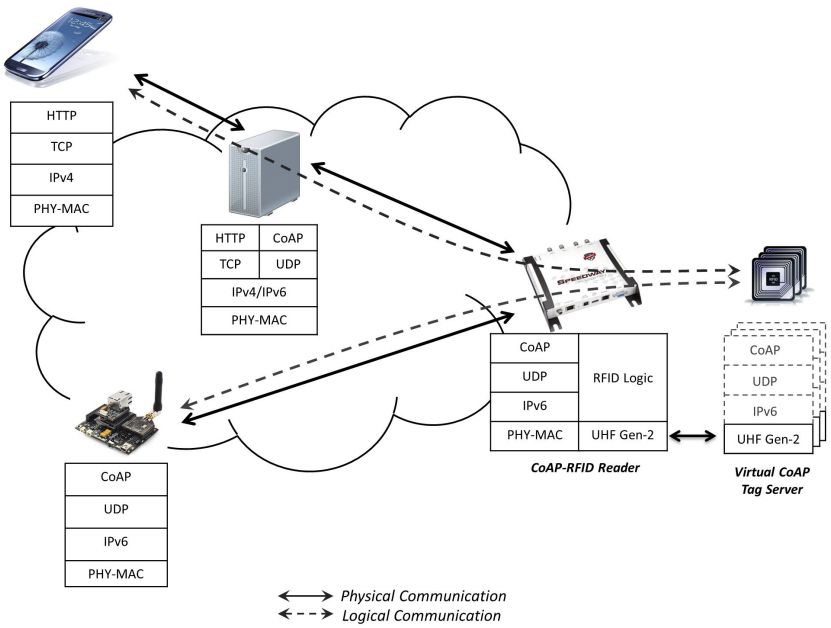


Figure 2. Protocol stack.

In the remainder of this section, we discuss the key design aspects of the "CoAP-RFID proxy reader". More details can be found in [12].

4.1. Tag Addressing and Discovery

According to the IoT vision, in our proposal, each object is uniquely addressable. In particular, the RFID reader proactively performs periodic Inventory operations to identify new tags in its field of view and to check the connectivity of already discovered tags. Thus, as soon as a new RFID transponder is detected, the reader creates an IPv6 address for this tag by utilizing the EPC code and its sub-network prefix, as in [39]. Then, the tag is registered in the RD, by using its EPC code as the endpoint name to facilitate future discovery operations.

In CoAP [11], the discovery procedure involves two operations: service and resource discovery. Service discovery allows one to obtain the entry point of a CoAP server, defined by the tuple {protocol, host, port}; resource discovery retrieves any feature or functionality that an endpoint offers to a remote client via REST-based interactions.

As for service discovery, in our solution, a client can obtain the IPv6 addresses of the RFID tags by querying the CoAP RD implemented on the RFID reader. The RD answers by providing the entry points of the tags operating as virtual CoAP servers. To enable global discovery, the architecture proposed in [40] for generic smart things can be exploited to make also tag resources remotely accessible.

Resource discovery is the second step. According to the standard [11], each CoAP server must foresee a defined relative URI "/.well-known/core" as a default entry point for requesting the list of links about its hosted resources. Therefore, after the tag's IP address has been discovered, the client can perform the resource discovery using a GET to "/.well-known/core" on the virtual CoAP server. The reader intercepts the request directed to the tag, deduces the EPC code through the EPC-IPv6 address mapping and returns the associated resources in the CoRE Link Format [34].

4.2. Tag Resource Representation and CoAP Operations

The offered resources depend on the tag type. If an identity tag is used for item identification in the EPCglobal platform, then its unique resource is the EPC code used by the CoAP client as a pointer to the object information stored in the EPCIS (Electronic Product Code Information Service) databases. In this regard, the work proposed in [41] to seamlessly integrate RFID information systems into the Web of Things, by designing a RESTful architecture for the EPCIS, is complementary to our approach. A CoAP client could use our solution to discover the EPCs of tags and then retrieve the associated information from the EPCIS of the manufacturer in a RESTful manner as in [41].

Otherwise, if the tag locally stores information on the attached object, then the data stored in its memory are remotely accessible. Regarding the way these resources are presented to external client applications (e.g., in binary format or using logical structures), it is preferable to use a high-level data description to provide meaningful information. This solution allows constrained clients to save energy, by avoiding data conversion. It also guarantees compatibility with the data structures already defined in RFID standards, such as ISO 15962 [42] and the EPCglobal Tag Data Standard [43], by providing relevant built-in support for encoding and decoding of the tag memory content. In particular, the use of Object Identifiers (OIDs) allows one to uniquely identify and access the stored data elements that have a specific meaning within given application domains.

In the case of sensor-equipped tags like the WISPs, an RFID reader should export also the number and the types of sensors integrated in the tags and the relative configurations, according to the relevant communication schema and data formats. In this regard, RFID standardization organizations have defined specific encoding and processing rules for sensor-equipped tags, such as ISO 24753 [44]. Furthermore, proprietary tag platforms could be supported by designing and deploying the appropriate protocol conversion modules, thus guaranteeing extendibility and flexibility. Furthermore, by abstracting the interaction between reader and tags over the RFID radio

interface, a common resource model could be provided to access sensor resources embedded in smart RFID platforms. To introduce the desired interoperability, remarkable efforts have been made to define coherent semantics for the Web of Things [45].

Once the resources exposed by a virtual tag are discovered, CoAP methods can be used to access them according to the RESTful guidelines. Each method transmitted in a CoAP request to the virtual RFID tag endpoint is mapped onto RFID commands from the RFID reader to the tags. Since data contained in the tag memory are considered as CoAP resources, these can be directly accessed by a GET operation. Analogously, a remote client can access the data collected by a sensor-equipped tag. Other CoAP operations, like POST or PUT, can be exploited to add new information into the tag memory and, in the case of sensors, to modify their settings; whereas DELETE can be used to reset the information contained in a tag. Obviously, a client must have the necessary rights to perform these operations.

CoAP supports advanced functionalities that enable interesting applications. Among them, through a publish-subscribe mechanism, observing resources avoids a continuous polling by a client to monitor the resource state. In particular, if a client inserts an observe option in the GET request, then the server will send a notification whenever the state of the interested resource changes. The observe option has been extended in [46] to integrate users' criteria within the observation request. In our scenario, the observation request directed to virtual servers is processed by the "CoAP-RFID proxy Reader", which periodically scans the content of the in-range tags and monitors the requested resources. As the burden of a continuous monitoring is transferred to the reader, conditional observe guarantees at least two advantages: (i) constrained nodes can save energy; and (ii) all of the observation requests can be managed in a centralized manner, thus optimizing operations on the radio interface. Obviously, there is an inherent trade-off between the frequency of observation and the update delay to inform client about resource changes.

5. Application Scenarios: An Example

The integration of the RFID technology in the IoT can bring benefits in several application scenarios, such as:

- Improved supply chain management: implements an exhaustive item-level Tracking, Tracing and Monitoring (TTM) systems, through the combined use of RFID for tracking and tracing, and sensors for monitoring perishable foods and drugs [47];
- Smart home: offers a ubiquitous home network system, where devices seamlessly embedded in objects of common use (such as sensors, actuators and tags) and Internet-based applications exchange information to enhance everyday life at home [48];
- Personal healthcare systems: track the human wellness and monitor the quality of the local environment to provide a remote patient monitoring and support system able to detect the presence of a person inside the room, his/her motion and interactions with nearby objects (such as medicines) [49].

Among these, of particular interest is the personal smart health scenario, since the possibility to monitor the user health and activate remote assistance is gaining more and more attention due to both the rise in life expectation and the availability of a wide variety of increasingly inexpensive sensors. In such a scenario, RFID systems represent a key enabler because of the energy-autonomy and the low cost of battery-less tags that make them compatible with a widespread distribution and disposable applications [49].

Figure 3 depicts a possible future scenario, wherein different kinds of resource-constrained nodes interact to implement a patient monitoring and support system. We envision different kinds of resource-constrained devices:

- Tag: an RFID tag with memory storage that acts as a patient's personal health profile information card, containing the values of the main vital signs and/or medical records;

- Sensor tag: an implantable RFID glucose-sensing microchip [50], able to continuously measure glucose levels in individuals with diabetes less invasively than traditional techniques;
- Sensor: a healthcare sensor interfaced to the pulse oximeter used to reliably assess patient vital health metrics, Heart Rate (HR) and blood oxygen saturation;
- Actuator: a wireless insulin pump that delivers insulin into the circulatory system of a diabetic patient's body.

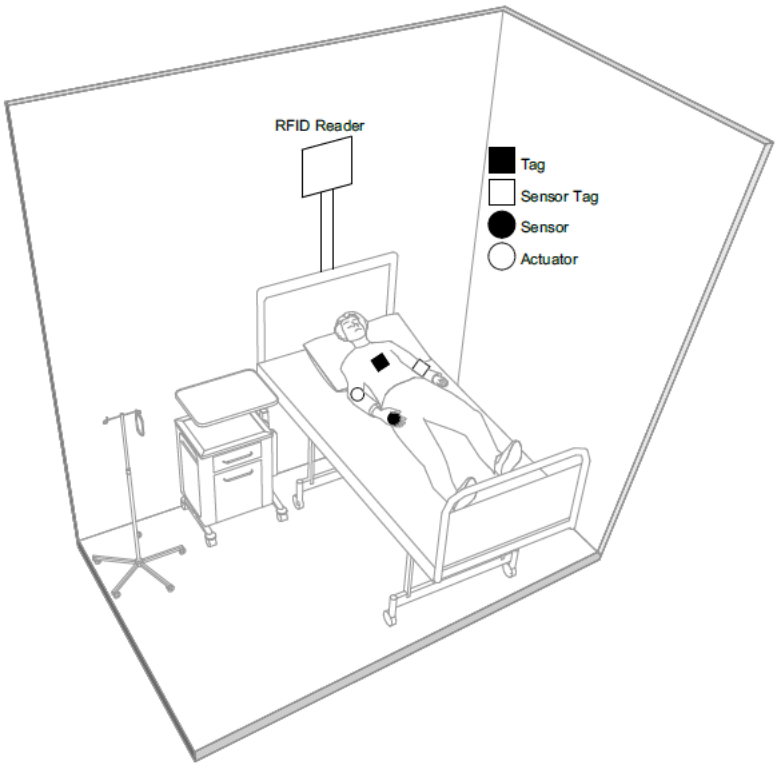


Figure 3. RFID-enabled IoT scenario.

In the considered scenario, we assume that a fixed RFID reader is installed in a hospital room and manages tags on the body of the patients located in the room. Nowadays, fixed readers typically support multiple antenna transmissions, so that a single device can cover a small environment, such as a room of a hospital with many patients' beds. Readers can be connected to the Internet/intranet and be accessible from clients (such as the smartphone or the desktop computer of a medical practitioner) using different connectivity means (e.g., Ethernet, Wi-Fi, cellular network). We consider this solution viable and cost efficient for a number of reasons. First, the continuous technological advancements in RFID technologies has been making the deployment cost of commercial RFID readers lower and lower. Second, using simple passive tags instead of more capable sensor nodes or sensor/actuator-equipped RFID tags to provide storage functionalities and record patient's personal health profile lowers manufacturing and maintenance costs due to the additional electronic components and periodic battery replacement that more advanced devices would require.

Clearly, readers still represent the main deployment cost of the proposed RFID system, and therefore, their positioning must be accurately planned, in order to guarantee an efficient tag reading while maintaining an affordable deployment cost. In addition, the efficient management of a networked infrastructure that connects RFID readers is essential to enable full access of tags' resources according to the application requirements. In the last few years, several middlewares have been developed to control the real-time operation of RFID readers, so as to efficiently set up the radio parameters based on the number and types of tags under their coverage, as well as to constantly monitor their status. In this way, the maintenance cost can be reduced, while enabling the realization

of an overall RFID system performing extremely well and reliably. For these reasons, we believe that RFID tags will play a key role in smart healthcare environments. In this regard, our integration solution could foster the widespread adoption of RFID systems, promoting the interoperability with a multitude of heterogeneous IoT devices.

In this scenario, two kinds of interactions take place: M2M (Machine to Machine) communications between resource-constrained nodes (even of different technologies) and remote real-time access to resources offered by constrained devices. Figure 4 depicts the different procedures when RFID devices are involved in the above-mentioned interactions.

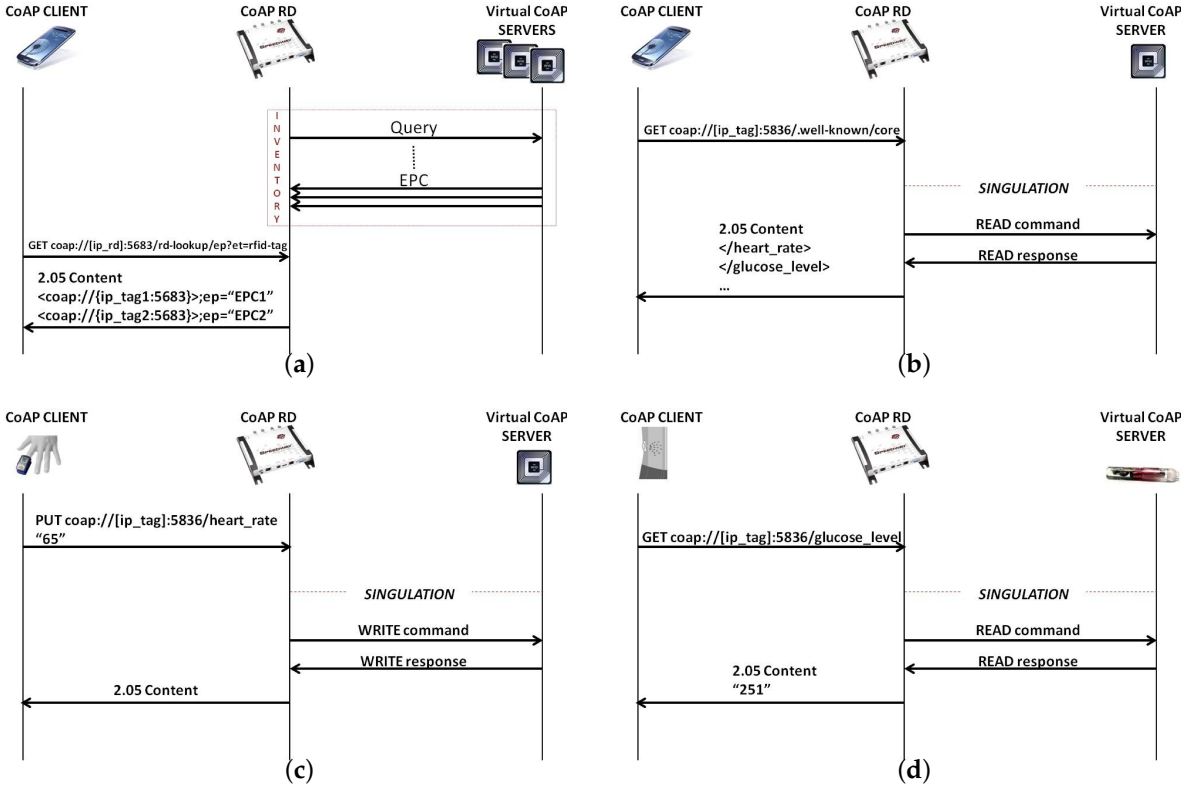


Figure 4. Message exchange. (**a**) Service discovery; (**b**) resource discovery; (**c**) sensor to tag communication; and (**d**) actuator to sensor tag communication.

A client (such as the smartphone of a medical practitioner) is interested in discovering all of the RFID devices in a certain location under the coverage area of an RFID reader. As shown in Figure 4a, the client will issue a GET request addressed to the RD implemented in our "augmented" reader, querying for all of the endpoints of type "rfid-tag" registered to the RD. Subsequently, the client checks what kind of information is stored in the personal health profile card of a certain patient (identified by an EPC code). In fact, the memory of passive RFID cards can be exploited to store remarkable information of the patient's medical history, as envisaged in [51]. To obtain the list of URLs about resources hosted by a tag, the remote client will submit a GET request, by specifying the default entry point "/.well-known/core", to the virtual CoAP server associated with the tag (see Figure 4b). After receiving the request, the reader will issue a Read command to the user memory of the specific RFID tag and parse contained resources in CoRE Link Format [34]. An appropriate ontology should be used to guarantee the semantic interoperability among the devices involved in the scenario, whereas the data format to store the information in the tag memory might either be raw text or adhere to tag data standards [43] and ISO 15962 [42] encoding schemes.

Information stored in each personal card also needs periodic updates. For example, the pulse oximeter must communicate with the patient's RFID card to write onto its user memory the last

measured value of heart beat. Figure 4c depicts this kind of M2M communication. In particular, the sensor will send a PUT request to the CoAP virtual tag, which will invoke a Write command on the tag's user memory by the reader.

In the near future, many operations that today require human intervention, such as the self-monitoring of blood glucose, will be transparently executed via device-to-device communication. This vision is considered as the next step to effectively boost the Web of Things. In our specific scenario, a wireless insulin pump could continuously check the glucose level measured by an implantable microchip embedded in the sensor tag and, if necessary, perform insulin injection, as shown in Figure 4d.

6. Implementation

We implemented the introduced platform by leveraging Californium [13], an open-source CoAP framework in Java. This platform is compliant with the CoAP standard [11] and provides additional useful features, such as the observe option and block-wise transfer. Moreover, it offers a convenient framework to implement server functionality.

Figure 5 illustrates the overall system architecture. First of all, we have extended Californium to support RFID operations, such as inventory, reading and writing. We provided a set of common APIs, defining a logical reader interface, that guarantee an abstraction layer from the communication interface of the RFID reader. In this way, our platform can be directly executed on a RFID reader (or on a server with a direct connection to the RFID reader), by implementing the appropriate protocol compliant with the logical reader interface. For instance, the RFID reader can be controlled by using the standardized LLRP protocol or vendor protocols.

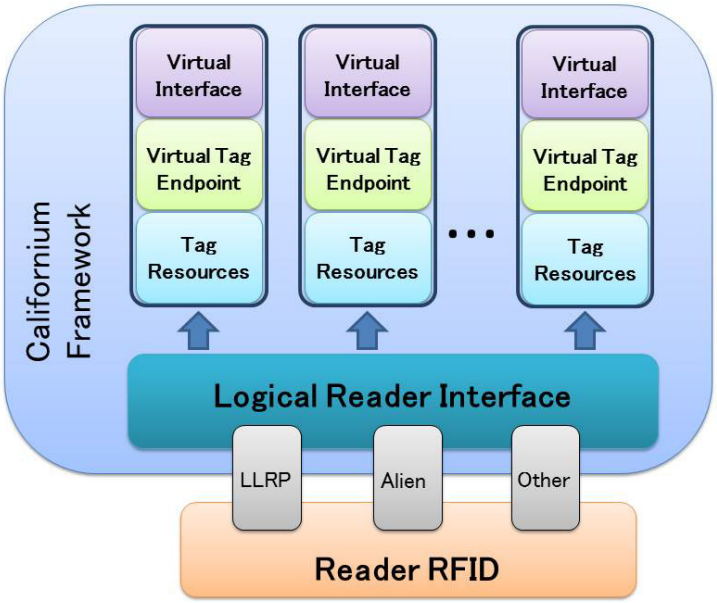


Figure 5. Architecture of the implemented "CoAP-RFID proxy reader".

The current version of Californium is not designed to support virtual endpoints. We recall that the proxy needs to create a specific virtual interface for each tag. To this aim, we have modified it to: (i) compute the IPv6 address of each new tag found during the periodic inventory from the tag identifier; (ii) generate its virtual interface; and (iii) identify the tag type to determine the tag virtual resources to be offered accordingly.

These resources can be exposed externally, by implementing the features of the resource directory entity. In particular, each virtual CoAP server presents its own tree of resources, which are managed consistently with the actual tag logic. The "CoAP-RFID proxy reader" has been designed in a flexible

and modular approach, so that it could be extended to support different data formats of legacy tags. We also underline that the proposed "CoAP-RFID proxy reader" is able to appropriately manage a high number of RFID tags under the reader's coverage, as this condition is likely in real IoT environments. Indeed, the provisioning of a virtual CoAP server for each tag just requires the creation of a new CoAP endpoint, which is associated with the tag address. Therefore, only one instance of the Californium platform is effectively running on the augmented RFID reader, and thus, the resource consumption of processing and memory resources is kept low also when a high number of tags is managed by the reader.

Different use cases require the deployment of an infrastructure composed of multiple readers, both fixed and mobile, to guarantee an appropriate coverage and to meet specific application requirements. Our solution can be adopted to provide access to tags' resources, even during short-lived interaction between mobile reader and tag. Indeed, if the reader is equipped with long-range communication radio interfaces, such as cellular networking, the RFID tags under its coverage could be exposed as virtual CoAP endpoints and remotely accessed.

A multi-reader environment introduces different challenges. First of all, particular attention should be addressed to the positioning of the readers, especially in indoor environments, to maximize the connectivity time of the tags deployed in the scenario, while minimizing the reader interferences and the deployment costs [52–54]. To guarantee fast service and resource discovery, the "CoAP-RFID proxy reader" operates as a resource directory. According to the application scenario, the resource directory features could be centralized, and the reader proactively manages the registration of a virtual CoAP server for each tag to the centralized resource directory. A different approach could foresee a distributed architecture, which uses a hierarchical approach or a peer-to-peer (P2P) solution to guarantee fast response times [55].

Complementary to this aspect, service continuity must be considered in a mobile environment in which the virtual CoAP server associated with a tag roams across areas controlled by different readers. Therefore, the system should not only provide networking connectivity, by adopting appropriate IPv6 mobility protocol (i.e., MobileIP or Locator Identity Separation Protocol - LISP), but it should also consider the context migration of the virtual CoAP server in order to complete on-going CoAP operations on the tag, such as updating of user memory values.

7. Testbed Setup and Experimental Results

In our testbed, the "CoAP-RFID proxy reader" is running on a 64-bit Linux OS laptop with an Intel Core i7 2630QM @ 2.20 GHz, 6 GB RAM and JavaSE-1.7. It is directly connected through an Ethernet wire to the RFID reader, which is an Alien ALR-9900 EMA+. The proprietary protocol, the Alien Reader Protocol (ARP), is used to issue commands to the reader. This implementation choice is motivated by the fact that commercial readers do not allow either for modifying the original firmware or for executing application code on top of the reader. From an architectural point of view, the laptop and the reader represents a single logical entity, the "CoAP-RFID proxy reader" previously described. The practical deployment of our proposal would only need a firmware update for the readers, since it does not require any hardware modification.

Moreover, we use UHF Impinj Monza tags, equipped with an EEPROM user memory of 512 bits. The CoAP client is running on a different workstation with features analogous to the other laptop, with which it communicates via a wireless IEEE 802.11 connection. The testbed is outlined in Figure 6. We highlight that the used commercial UHF RFID tags present high reading and writing sensitivity, which allows for transmissions up to 5 to 10 m according to environmental conditions.

The objectives of the performance assessment study are to evaluate the feasibility of the proposed approach and to provide useful insights for the design of CoAP-compliant integration solutions for RFID systems. In our implementation, particular attention to the management of the tag resources is deserved, which can be directly accessed to perform appropriate reading and writing operations on the tag memory. Indeed, a key aspect of our proposal is the access to a common set of RFID

resources through standardized IETF IoT protocols, similarly to the way resources of other smart things are accessed. The ultimate aim is to facilitate and foster the deployment of interoperable RFID applications in the IoT domain. To demonstrate how our solution can effectively ease the management of RFID tags, we use the graphical CoAP client Copper [13], which allows developers to test device APIs in a user-friendly manner. Figure 7a reports a screenshot showing the result of a GET request to the EPC code resource of a tag in the field of view of the reader. For the sake of simplicity, the EPC code is provided in hexadecimal format, but the application logic can decide the most suitable representation according to the Tag Data Standard [43]. In Figure 7b, we show the result of a GET request to the user memory resource, whose content is presented in hexadecimal format. Furthermore, the Copper interface allows one to graphically investigate the main fields of the messages involved in the CoAP request-response interaction.

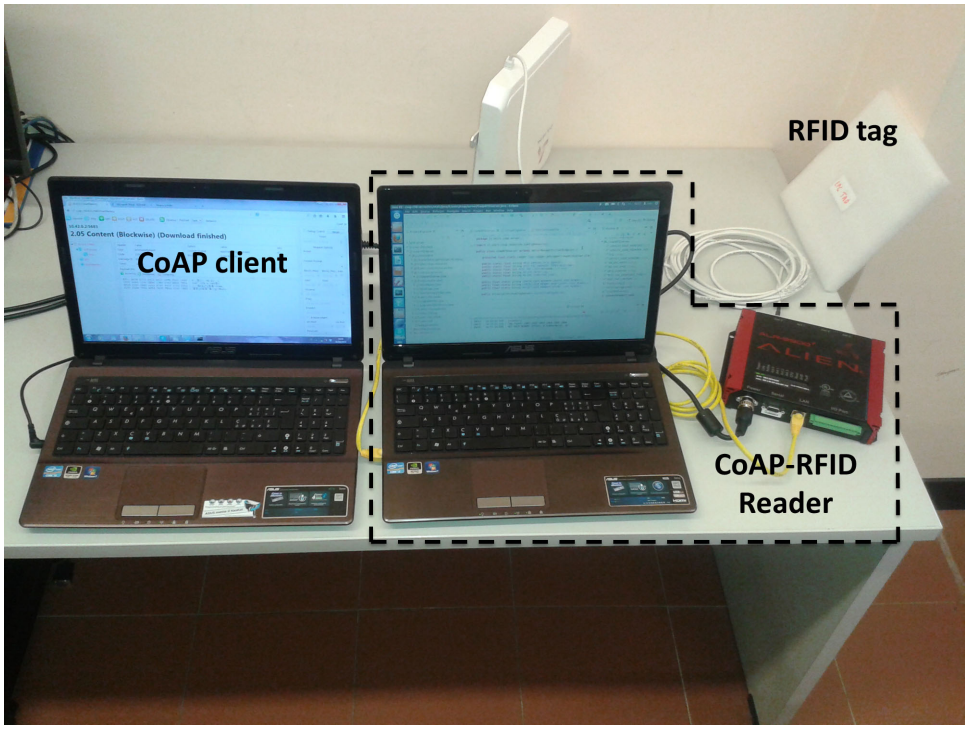


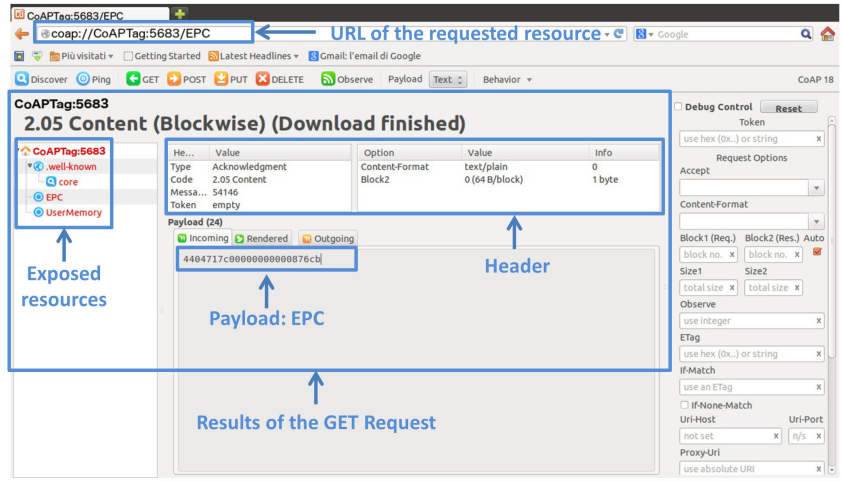
Figure 6. Testbed environment.

To evaluate how RFID features influence CoAP-based communications, we measure the values of parameters strongly influencing energy consumptions of low-power devices (as this feature plays a central role in IoT scenarios). The following metrics are considered:

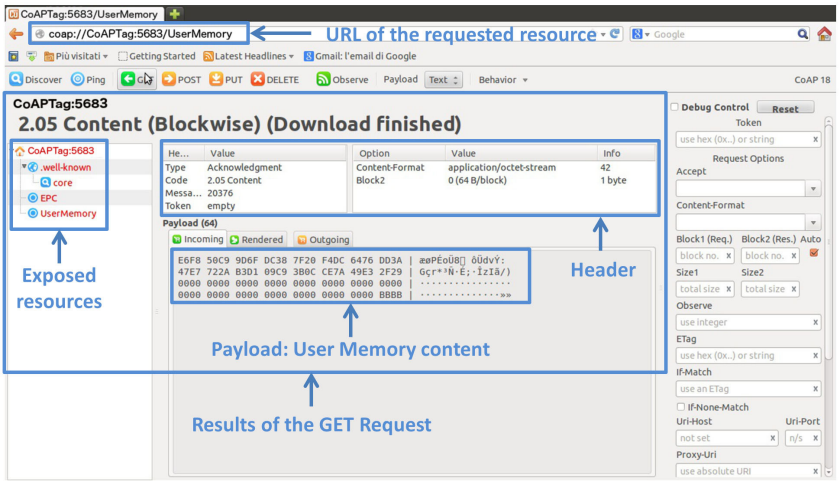
- the execution time required by the CoAP proxy to satisfy a specific request;
- the total number of transferred bytes for the CoAP request and response packets required to access the tag resources.

The parameters measure the responsiveness of the proposed approach and the amount of data that needs to be exchanged in a real environment, such as the personal smart health scenario discussed in the previous section.

In our experimental set-up, a CoAP client interacts with a tag, which acts as a virtual CoAP server, to retrieve a resource (GET method) and to modify an existing resource (PUT method). In particular, we analyze both the execution time and the packets size for GET and PUT CoAP methods of the user memory, whose size varies from 128 to 512 bits.



(**a**)



(**b**)

Figure 7. Screenshots of Copper showing the result of a GET request to: (**a**) the EPC code; (**b**) the user memory.

In Figure 8, we plot the execution time for GET and PUT requests. The response time of the reader significantly influences the overall system performance. The Alien reader uses a pre-defined set of Gen2 air protocol parameters [14], where the Tari value is set to 25 µs and Miller encoding is used. The resulting reader-to-tag and tag-to-reader rates are equal to 29 and 250 kbps, respectively. The power transmission of the reader is set to the default value of 2 W, whereas the distance between the reader's antenna and the tag is around 1.5 m in our testbed environment. The GET response times require around 60 ms to retrieve the whole user memory, whereas PUT response times vary from 80 to 360 ms for the considered user memory sizes. The PUT method is more time consuming since a reader can retrieve the content of the whole user memory by using a single Read command, whereas writing operations can be performed by only processing blocks of 16 bits at a time. Thus, the reader needs to send several Write commands to update the whole memory bank. Furthermore, the writing operation is strongly influenced by the tag memory performance, which typically requires 4 to 10 ms for each 16-bit word update in case of EEPROM memory. Accounting for the time required to access the resources offered by a single tag, appropriate strategies could be adopted in the case of multiple tags under the coverage of the same reader and parallel requests. Indeed, we underline that the reader could operate on only one tag at a time over the UHF RFID radio interface. Therefore, when multiple CoAP requests are received, the execution time depends both on the elaboration time of the single operation and the queueing delay before the effective execution of the operation itself.

An appropriate scheduling of the requests must be designed, according to their priority and to the workload. Furthermore, the CoAP protocol may implement some features in case the server causes long delays in executing a request, in order to avoid repeated client retransmissions due to missed server response within a given time interval. When a client requires retrieving/modifying a resource through the delivery of a confirmable message, the server can answer with either a piggy-backed or a separate response. In the former case, the response is included in a CoAP message that acknowledges (ACK) the request reception. If the server is unable to respond immediately to a request carried in a confirmable message, then it can simply respond with an empty ACK message (this prevents the client from repeatedly retransmitting the request), and a response is then sent in a separate message exchange. As specified in the CoAP standard, the parameter that guides the decision of adopting either a piggy-backed or a separate response is the processing delay (default value is 2 s), defined as the time a node should take to acknowledge a confirmable message. Therefore, the "CoAP-RFID proxy reader", on behalf of the relevant virtual CoAP servers, could implement separate responses for the received requests whose estimated execution times overcome the threshold of the processing delay.

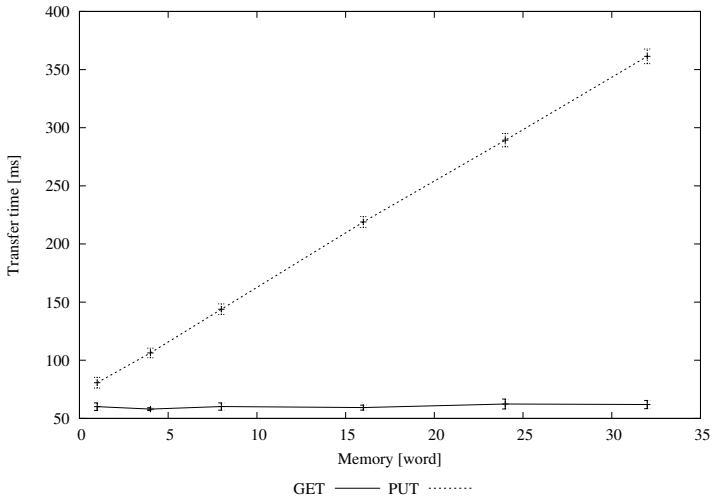


Figure 8. Execution time.

Finally, in Figure 9, we plot the size of the packets at the CoAP level for GET and PUT operations, explicitly reporting the details of both request and response messages. The figures highlight that the total amount of transferred bytes is similar in the case of the GET and PUT operations, but Request and Response messages have different sizes in the two cases. In particular, the dimension of the Request message for a GET operation is constant (it contains only the header), while the size of the Response message increases when the dimension of the user memory grows (it carries the content of the user memory). These considerations can be inferred by looking at Figure 9a. In the case of a PUT operation (see Figure 9b), we have the opposite situation, since the value of the resource to be modified is transferred in the Request message while no payload is carried in the Response message. The total number of transferred bytes highly influences the power consumption of resource-constrained nodes. Normally, link-layer protocols, such as IEEE 802.15.4, impose a tight limit in the dimension of the data payload, so that large-sized application messages are fragmented by the appropriate adaptation layer, e.g., 6LoWPAN, before being transmitted over the air interface. In the case of channel errors, the loss of a single fragment could imply the retransmission of the whole original packet. To avoid this, the CoAP protocol offers the so-called block-wise transfer option, which allows for performing large message segmentation into multiple packets directly at the application layer (rather than 6LoWPAN fragmentation). In this regard, information on the request/response message size could be useful to guide the decision of the CoAP server on the adoption of the block-wise option to maximize the probability of packet delivery for large payloads.

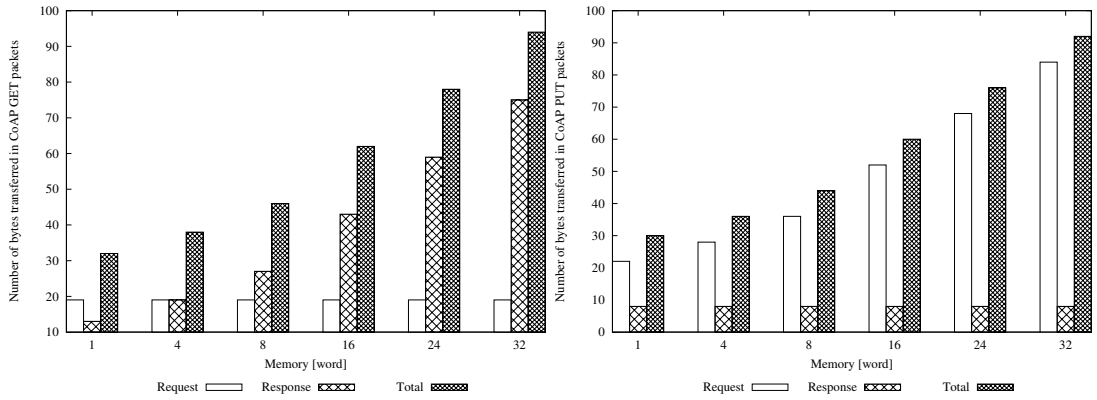


Figure 9. Amount of transferred bytes in CoAP: (**a**) GET; and (**b**) PUT packets.

8. Conclusions

In this paper, we have thoroughly discussed the issue of an efficient inclusion of the RFID technology in the IoT, by analyzing the pro and cons of different integration schemes. Accounting for the current capabilities of RFID technology, we have proposed a CoAP-compliant solution that allows one to directly access RFID tags (and related resources) via the reader. In the conceived architecture, the RFID reader works as a proxy for the managed tags, which act as virtual CoAP servers. We have accurately investigated the benefits of exploiting the proposed solution in a promising healthcare environment, by evaluating the interactions among smart health devices in such an environment. Then, we have implemented our proposal, by leveraging an open-source CoAP framework, and we have assessed the performance in a real scenario.

Author Contributions: I.F. and S.P. conceived the idea, designed, and implemented the proposed solutions, performed the experiments; all the authors analyzed the data and wrote the paper; furthermore, all the authors reviewed the writing of the paper, its structure and its intellectual content.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Smith, J.R.; Sample, A.P.; Powledge, P.S.; Roy, S.; Mamishev, A. A wirelessly-powered platform for sensing and computation. In *UbiComp 2006: Ubiquitous Computing, Lecture Notes in Computer Science*; Springer: Orange County, CA, USA, 2006; Volume 4206, pp. 495–506.
2. De Donno, D.; Catarinucci, L.; Tarricone, L. A battery-assisted sensor-enhanced RFID tag enabling heterogeneous wireless sensor networks. *IEEE Sens. J.* **2014**, *14*, 1048–1055.
3. Khan, M.S.; Islam, M.S.; Hai, D. Design of a reconfigurable RFID sensing tag as a generic sensing platform toward the future Internet of Things. *IEEE Internet Things J.* **2014**, *1*, 300–310.
4. Buettner, K.; Greenstein, B.; Sample, A.; Smith, J.R.; Wetherall, D. Revisiting smart dust with RFID sensor networks. In Proceedings of the 7th ACM Workshop on Hot Topics in Networks (HotNets-VII), Calgary, AB, Canada, 6–7 October 2008.
5. Jara, A.J.; Zamora, M.A.; Skarmeta, A.F. An Internet of Things-based personal device for diabetes therapy management in ambient assisted living (AAL). *Pers. Ubiquitous Comput.* **2011**, *15*, 431–440.
6. Yeager, D.J.; Powledge, P.S.; Prasad, R.; Wetherall, D.; Smith, J.R. Wirelessly-charged UHF tags for sensor data collection. In Proceedings of the IEEE International Conference on RFID, Las Vegas, NV, USA, 16–17 April 2008; pp. 320–327.
7. Merenda, M.; Farris, I.; Felini, C.; Militano, L.; Spinella, S.C.; Della Corte, F.G.; Iera, A. Performance assessment of an enhanced RFID sensor tag for long-run sensing applications. In Proceedings of the IEEE Sensor, Valencia, Spain, 2–5 November 2014; pp. 738–741.
8. Ruiz-Garcia, L.; Lunadei, L.; Barreiro, P.; Robla, I. A review of wireless sensor technologies and applications in agriculture and food industry: State of the art and current trends. *Sensors* **2009**, *9*, 4728–4750.

9. Farris, I.; Militano, L.; Iera, A.; Molinaro, A; Spinella, S.C. Tag-based cooperative data gathering and energy recharging in wide area RFID sensor networks. *Ad Hoc Netw.* **2016**, *36*, 214–228.
10. Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805.
11. Shelby, Z.; Hartke, K.; Bormann, C.; Frank, B. The Constrained Application Protocol (CoAP); RFC 7252; IETF, 2014. Available online: https://www.rfc-editor.org/info/rfc7252 (accessed on 26 October 2016).
12. Farris, I.; Iera, A.; Molinaro, A.; Pizzi, S. A CoAP-compliant solution for efficient inclusion of RFID in the Internet of Things. In Proceedings of the IEEE Globecom, Austin, TX, USA, 8–12 December 2014; pp. 2795–2800.
13. Kovatsch, M.; Mayer, S.; Ostermaier, B. Moving application logic from the firmware to the cloud: Towards the thin server architecture for the Internet of Things. In Proceedings of the IEEE International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), Palermo, Italy, 4–6 July 2012; pp. 751–756.
14. *Specification for RFID Air Interface, EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz–960 MHz*, Version 2.0.0; EPC Global: Lawrenceville, NJ, USA, 2013.
15. *ISO/IEC 18000-63—Information Technology—Radio Frequency Identification for Item Management—Part 63: Parameters for Air Interface Communications at 860 MHz to 960 MHz Type C*; ISO: Geneva, Switzerland, 2013.
16. *Application Level Events (ALE)*, Version 1.1.1; EPCGlobal Specification; EPC Global: Lawrenceville, NJ, USA, 2009.
17. *Low Level Reader Protocol (LLRP)*, Version 1.1; EPCGlobal Specification; EPC Global: Lawrenceville, NJ, USA, 2010.
18. Jara, A.J.; Moreno-Sanchez, P.; Skarmeta, A.F.G.; Varakliotis, S.; Kirstein, P. IPv6 Addressing Proxy: Mapping Native Addressing from Legacy Technologies and Devices to the Internet of Things (IPv6). *Sensors* **2013**, *13*, 6687–6712.
19. IoT6 Project. Available online: http://www.iot6.eu/ (accessed on 26 October 2016).
20. EPCglobal. Available online: http://www.gs1.org/epcglobal (accessed on 26 October 2016).
21. Farris, I.; Iera, A.; Molinaro, A.; Pizzi, S. A Novel IPv6-based Approach to Exploit the Potentials of UHF RFID for Smart Factory 4.0. Available online: http://mmc.committees.comsoc.org/files/2016/04/E-Letter-September2015.pdf#page=24 (accessed on 26 October 2016).
22. IETF IPv6 Over Networks of Resource-Constrained Nodes (6lo) Working Group. Available online: https://tools.ietf.org/wg/6lo (accessed on 26 October 2016).
23. Nieminen, J.; Gomez, C.; Isomaki, M.; Savolainen, T.; Patil, B.; Shelby, Z.; Xi, M.; Oller, J. Networking solutions for connecting bluetooth low energy enabled machines to the Internet of Things. *IEEE Netw.* **2014**, 28, 83–90.
24. Transmission of IPv6 Packets over DECT Ultra Low Energy; draft-ietf-6lo-dect-ule-06. Available online: https://tools.ietf.org/html/draft-ietf-6lo-dect-ule-06 (accessed on 26 October 2016).
25. Transmission of IPv6 Packets over Near Field Communication; draft-ietf-6lo-nfc-04. Available online: https://tools.ietf.org/html/draft-ietf-6lo-nfc-04 (accessed on 26 October 2016).
26. Liu, A.; Hu, Y.; Chen, Z. An energy-efficient mobile target detection scheme with adjustable duty cycles in wireless sensor networks. *Int. J. Ad Hoc Ubiquitous Comput.* **2016**, *22*, 203–225.
27. Liu, Y.; Dong, M.; Ota, K.; Liu, A. ActiveTrust: Secure and trustable routing in wireless sensor networks. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 2013–2027.
28. He, S.; Chen, J.; Li, X.; Shen, X.S.; Sun, Y. Mobility and intruder prior information improving the barrier coverage of sparse sensor networks. *IEEE Trans. Mob. Comput.* **2014**, *13*, 1268–1282.
29. Farris, I.; Felini, C.; Pizzi, S.; Merenda, M.; Iera, A.; Della Corte, F.; Molinaro, A. Enabling communication among smart tags in an UHF RFID Local Area Network Internet of Things (WF-IoT). In Proceedings of the IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, Italy, 14–15 December 2015; pp. 524–529.
30. Farris, I.; Iera, A.; Spinella, S.C. Introducing a Novel "Virtual Communication Channel" into RFID Ecosystems for IoT. *IEEE Commun. Lett.* **2013**, *17*, 1532–1535.
31. Farris, I.; Militano, L.; Iera, A.; Spinella, S.C. Assessing the performance of a novel tag-based reader-to-reader communication paradigm under noisy channel conditions. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 4813–4825.

32. Fielding, R.T. Architectural Styles and the Design of Network-based Software Architectures. Ph.D. Thesis, University of California, Irvine, CA, USA, 2000.
33. Shelby, Z.; Krco, S.; Bormann, C. *CoRE Resource Directory*; Proposed Standard; IETF: Fremont, CA, USA, 2016.
34. Shelby, Z. *Constrained RESTful Environments (CoRE) Link Format*; *RFC 6690*; IETF: Fremont, CA, USA, 2012.
35. Sung, J.; Lopez, T.S.; Kim, D. The EPC sensor network for RFID and WSN integration infrastructure. In Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, New York, NY, USA, 19–23 March 2007; pp. 618–621.
36. Tseng, C.W.; Chang, C.M.; Huang, C.H. Complex sensing event process of IoT application based on EPCglobal architecture and IEEE 1451. In Proceedings of the International Conference on the Internet of Things (IOT), Wuxi, China, 24–26 October 2012; pp. 92–98.
37. Lynn, K.; Shelby, Z. *CoRE Link-Format to DNS-Based Service Discovery Mapping*; draft-lynn-core-discovery-mapping-02; IETF: Fremont, CA, USA; 2012.
38. Castellani, A.; Loreto, S.; Rahman, A.; Fossati, T.; Dijk, E. *Guidelines for HTTP-to-CoAP Mapping Implementations*; draft-castellani-core-http-mapping-10; IETF: Fremont, CA, USA, 2016.
39. Jensen, S.E.H.; Jacobsen, R.H. Integrating RFID with IP host identities. In *Radio Frequency Identification from System to Applications*; InTech: Rijeka, Croatia, 2013.
40. Lopez, P.; Fernandez, D.; Marin-Perez, R.; Jara, A.J.; Gomez-Skarmeta, A.F. Scalable Oriented-Service Architecture for Heterogeneous and Ubiquitous IoT Domains. *Pervasive and Mobile Computing*. Available online: http://arxiv.org/abs/1311.4293 (accessed on 26 October 2016).
41. Guinard, D.M.; Mueller, M.; Trifa, V. RESTifying real-world systems: A practical case study in RFID. In *REST: From Research to Practice*; Springer: Berlin, Germany, 2011; pp. 359–379.
42. *ISO/IEC 15962–Information Technology—Radio Frequency Identification (RFID) for Item Management—Data Protocol: Data Encoding Rules and Logical Memory Functions*; ISO: Geneva, Switzerland, 2013.
43. *EPC Tag Data Standard*, Version 1.8; GS1 Standard: Lawrenceville, NJ, USA, 2014.
44. *ISO/IEC 24753—Information Technology—Radio Frequency Identification (RFID) for Item Management—Application Protocol: Encoding and Processing Rules for Sensors and Batteries*; ISO: Geneva, Switzerland, 2011.
45. Jara, A.J.; Olivieri, A.C.; Bocchi, Y.; Jung, M.; Kastner, W.; Skarmeta, A.F. Semantic web of things: An analysis of the application semantics for the IoT moving towards the IoT convergence. *Int. J. Web Grid Serv.* **2014**, *10*, 244–272.
46. Ketema, G.; Hoebeke, J.; Moerman, I.; Demeester, P.; Tao, L.S.; Jara, A.J. Efficiently observing Internet of Things resources. In Proceedings of the IEEE International Conference on Green Computing and Commun (GreenCom), Besancon, France, 11–14 Septermber 2012; pp. 446–449.
47. Castro, M.; Jara, A.J.; Skarmeta, A. Architecture for improving terrestrial logistics based on the web of things. *Sensors* **2012**, *12*, 6538–6575.
48. Perera, C.; Zaslavsky, A.; Christen, P.; Georgakopoulos, D. Sensing as a service model for smart cities supported by Internet of Things. *Trans. Emerg. Telecommun. Technol.* **2014**, *25*, 81–93.
49. Amendola, S.; Lodato, R.; Manzari, S.; Occhiuzzi, C.; Marrocco, G. RFID technology for IoT-based personal healthcare in smart spaces. *IEEE Internet Things J.* **2014**, *1*, 2327–4662.
50. Xiao, Z.; Tan, X.; Chen, X.; Chen, S.; Zhang, Z.; Zhang, H.; Wang, J.; Huang, Y.; Zhang, P.; Zheng, L.; et al. An implantable RFID sensor tag toward continuous glucose monitoring. *IEEE J. Biomed. Health Inf.* **2015**, *19*, 910–919.
51. Hawrylak, P.J.; Hart, C. Using radio frequency identification technology to store patients' medical information. In *Handbook of Research on Patient Safety and Quality Care through Health Informatics*; IGI Global: Hershey, PA, USA, 2014.
52. He, S.; Chen, J.; Jiang, F.; Yau, D.K.Y.; Xing G.; Sun, Y. Energy provisioning in wireless rechargeable sensor networks. *IEEE Trans. Mob. Comput.* **2013**, *12*, 1931–1942.
53. Yang, Q.; He, S.; Li, J.; Chen, J.; Sun, Y. Energy-efficient probabilistic area coverage in wireless sensor networks. *IEEE Trans. Veh. Technol.* **2015**, *64*, 367–377.

54. Shih, D.H.; Sun, P.L.; Yen, D.C.; Huang, S.M. Taxonomy and survey of RFID anti-collision protocols. *Comput. Commun.* **2006**, *29*, 2150–2166.
55. Villaverde, B.C.; Alberola, R.D.P.; Jara, A.J.; Fedor, S.; Das, S.K.; Pesch, D. Service discovery protocols for constrained machine-to-machine communications. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 41–60.