Pers Ubiquit Comput (2014) 18:323–338
DOI 10.1007/s00779-013-0648-0

ORIGINAL ARTICLE

Mobile digcovery: discovering and interacting with the world through the Internet of things

Antonio J. Jara · Pablo Lopez · David Fernandez · Jose F. Castillo · Miguel A. Zamora · Antonio F. Skarmeta

Received: 31 October 2012 / Accepted: 31 January 2013 / Published online: 10 March 2013

Abstract The application of Internet-enabled devices in the real world for the development of Smart Cities, environmental monitoring, bus tracking, and parking requires scalability, extensibility, and integration of emerging resources to reach a suitable ecosystem for data acquisition and interaction with citizens. Internet of things needs to offer efficient support for global communications and access to services and information. It needs to enable homogeneous and seamless machine-to-machine communication for different solutions and applications. This work presents an homogeneous and suitable mechanism for global resource discovery, device access for deployed smart objects in different scenarios, and sensors and devices from end users (participative sensing). The integration of legacy and sensors already available from smart buildings and smart objects is presented. For this purpose, a resolution infrastructure called "digcovery" is defined for maximizing efficiency and sustainability of deployments. Digcovery architecture offers the framework to allow users to register/include their own sensors into a common infrastructure and access/discover the available resources through mobile digcovery. Mobile digcovery exploits the context-awareness, geo-location, and identification technologies available in mobile platforms such as smartphones to discover, interact, and access the resources through its ElasticSearch engine.

Keywords Internet of things · Discovery · Governance · Context-awareness · Identification

A. J. Jara (✉) · P. Lopez · D. Fernandez · J. F. Castillo · M. A. Zamora · A. F. Skarmeta
Clinical Technology Lab (CliTech), Computer Science Faculty, University of Murcia, Murcia, Spain
e-mail: jara@um.es

P. Lopez
e-mail: p.lopezmartinez@um.es

D. Fernandez
e-mail: david.f.r@um.es

J. F. Castillo
e-mail: josefelix.castillo@um.es

M. A. Zamora
e-mail: mzamora@um.es

A. F. Skarmeta
e-mail: skarmeta@um.es

1 Introduction

The future is unpredictable, but the present is powered by a new generation of solutions and services based on higher context-awareness through geo-location and identification technologies. These capabilities are found in multiple applications developed for platforms such as tables and smartphones. The way that people interact with the world has changed, senses have been extended with the new generation of technologies, devices, and networks that are powered by a world of ubiquitous computing applications. These applications have made it feasible to provide and deliver information in any place around the world [1], with just a click on a smartphone screen. This new generation of services and interaction with the world through the smartphone has created a "smartphone generation" [2].

The smartphone generation is accustomed to the connected world, for them it is the norm to be connected to any person, anytime, anywhere. In fact, people increasingly want to be able to monitor and control every "thing" in their lives anytime, anywhere, and subsequently, his new connected world is stimulating consumer interest in

Internet-connected products [2]. Therefore, it is the long-standing vision of ubiquitous computing to provide this new connected generation everyday smart objects smart endowed with computing and communication capabilities. The idea of connecting things to the Internet has been around since its commencement and stems from the desire to remotely monitor and control objects such as the drinks vending machine connected to the Internet in the 1970s at the School of Computer Science in the Carnegie Mellon University [3].

Nowadays, this vision of a world with everything connected is not only limited to specific devices or anecdotic machines. The Internet is being extended to connect all objects and devices that surround us, this is the so-called Internet of things (IoT) [4]. The objective of IoT is to allow systems total control and access to other systems, which will result in the provision of an ubiquitous communication and computing facility. Hence, a new generation of smart and small devices, context-awareness services, and applications can be defined. IoT is currently applied in a wide number of areas such as home automation, advanced metering, smart grid, lighting, traffic management, and environment monitoring. All these application areas are now being aggregated into the concept of Smart Cities.

Consequently, Smart Cities are being composed of a large and heterogeneous number of devices, sensors, and actuators. This aggregation offers higher level solutions based on context-awareness through, on the one hand, tracking and location of buses, ambulances, and flows of people, and on the other, smart sensing of the different parameters such as traffic, parking, and environmental conditions (temperature, humidity, and luminosity). These higher level solutions allow smart performance through light intensity regulators, parking space detectors, and audiovisual displays for real-time and dynamic traffic management. These are only some examples of the latest solutions associated with the conception of Smart Cities.

Smart cities are a major advance in the conception of ubiquitous computing and the communications evolution. The number and diversity of sensors and devices deployed is growing tremendously thanks to their capacity to offer low-cost air interfaces which allow easy and quick deployment. It is also growing due to infrastructure capacity to provide Internet access to these networks and suitability to support an extended range of solutions. Consequently, Internet of things is becoming ubiquitous to all environments and users, and accessible for the sensors with the evolution of technologies such as IPv6 low-power wireless personal area networks (6LoWPAN). There has been a tremendous increase in Internet use since 2000, from 360 million users to 1.6 billion. There are 4 billion mobile users and over 570 million users of Internet-enabled handheld devices. The number of users further increases with the introduction of smart things to the Internet. It is estimated, in relation to the Internet of things, that the number of devices connected to the Internet will connect 50–100 billion [5] by 2020. Smart Cities can be considered the major test bed of the Internet of things and consequently its major challenge. The challenges arise as a consequence of this increase in the number of devices in terms of scalability, governance, security, and discoverability.

Since the users are part of these Smart Cities, how to deliver suitable user interaction, information collectiont and active participation needs to be solved. These are still open questions that need to be answered. This work is focused on the discoverability of devices based on context-awareness and geo-location. First, the building of a scalable architecture to support an extensible number of devices, domains, and the integration of heterogeneous technologies including radio frequency identification (RFID), sensor networks, near-field communication (NFC), Bluetooth, Zigbee, WiFi, and legacy technologies has been required. Second, an application to interact with the world through wide area capable devices and applications to discover services anywhere in the world has been built. Finally, the search engine, which is based on an elastic approach in order to suitably integrate distributed resources repositories and offer support for the discovery, i.e., query, lookup, and filtering, of resources based on context-awareness, resource types, and geo-location criteria, is presented.

Service discoverability is a design principle, applied within the service-orientation design paradigm, which emphasizes the ability to make services discoverable by adding interpretable meta-data to increase service reuse and decrease the chance of developing services that overlap in function. By making services easily discoverable, this design principle indirectly makes services more interoperable.

This work presents the mobile digcovery architecture to enable consumers to monitor and control their “things” (products) from web browsers and smartphones. Services and resources discovery has been discussed in the literature [6], but an approach for the integration of the different resources available in the Internet of things ecosystems such as Smart Cities has not yet been proposed, neither one that would offer suitable interaction through mobile platforms powered with a elastic and scalable search engine, such as that presented in Sect. 5, which is suitable for the discovery of heterogeneous and distributed resources. Digcovery architecture is presented in Sect. 2. This offers the infrastructure to carry out global discovery, integration of heterogeneous types of resources, and elastic capabilities to make the growth of the architecture scalable, in addition to the lookup and browsing of devices, services, and resources. Digcovery is seen as a middleware or mash-up enabler, since this integrated the devices, services, and resources from a wide range of legacy technologies, smart objects (IP-enabled), and Web Services. Digcovery architecture is funded over a collection of digrectories. Digrectory transforms any kind of device, service, and resource into a semantic interoperable format that allows them to be accessed, combined, and exploited by any application. Digrectory can also be seen as a connector, following the idea of the *Presto Mash-up Connectors* [7].

Mobile digcovery is the front end of the digcovery architecture; this offers capabilities to interact with the world through, on the one hand, the identification of the objects via technologies such as barcode, quick response (QR) code, and RFID/NFC tags, and on the other hand, the filtering of services based on geo-location and context-awareness. Section 4 presents the different mechanism for interaction offered by mobile digcovery, and Sect. 5 presents the search engine to filter by geo-location, context-awareness, and the resource types that the user is interested in.

Section 6 presents the other solutions defined for bus tracking, mash-up, and interaction with the real world through the smartphone. Section 7 presents the main differences and advantages of mobile digcovery with respect to the state of the art, and finally Sect. 8 concludes the paper.

2 Architecture

Digcovery architecture is presented in Fig. 1. This presents how the different technologies involved in the Internet of things ecosystem such as smart objects, RFID tags, and legacy devices are integrated into different digrectories.

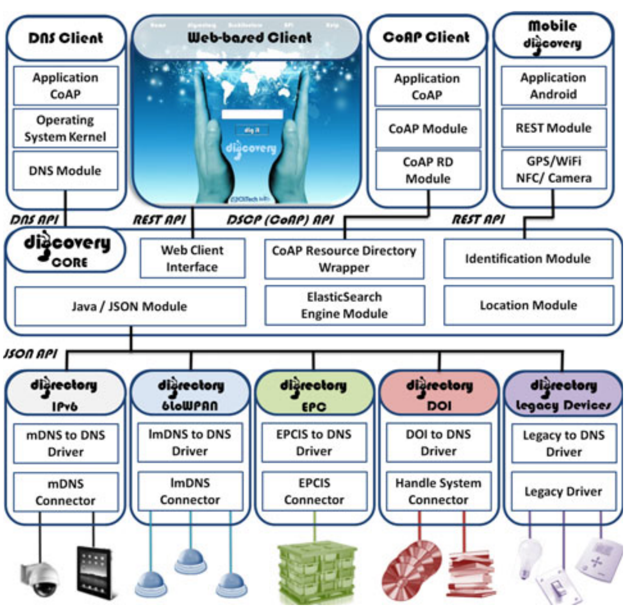

Fig. 1 Digcovery architecture

These digrectories are managed through DNS queries extended with an ElasticSearch engine in order to make it scalable; at the same time, it offers a centralized point, called digcovery core, to manage and discover them.

All the resources and services are mapped to a common ontology and description, based on existing ontologies and profiles from the IP-enabled smart objects alliance (IPSO) [8, 9] and the open mobile alliance (OMA) [10], and which is compatible with DNS-SD types, in order to reach a common semantic description accessible through DNS.

This also presents how to interoperate with the discovery architecture through other interfaces different to DNS such as RESTful architecture with data structured in JSON format. JSON has been chosen for the interoperability with all the resources, since it is considered by the Working Groups from the IETF such as the Constrained Resources (CoRE) Working Group and the Constrained Management (COMA) Working Group as the most suitable protocol to structure the data for constrained resources, leaving other formats such as XML optional. The lookup and queries (ElasticSearch) over digcovery with context-awareness, based on location or resource types, over the proposed ElasticSearch architecture, offers organized and context-based queries over a heterogeneous and distributed source of resources and services.

The first top module from Fig. 1 shows that the usage of the platform can be through DNS in order to exploit existing IP-based technologies, protocols, and mechanisms. The second top module is the web-based platform to access and register resources through the RESTFul architecture. Regarding the integration with the RESTFul architecture, this also offers an interface based on CoAP [11], the Digcovery CoAP Service Protocol (DCSP) for CoAP-enabled devices.

CoAP is the main result from the CoRE Working Group. This offers an integration of the RESTFul architecture for constrained devices with an overhead of only 4 bytes and a functionality optimized for the observation of resources [12], application layer fragmentation [13], and mapping with the HTTP-based RESTFul architecture.

The final module supported by the digcovery architecture is the mobile digcovery, which extends the architecture through the identification and location capabilities offered by the mobile platforms such as smartphones through integrated technologies, i.e., global position systems (GPS) and WiFi (real-time location systems) for location, and RFID and cameras (barcodes and QR codes) for identification. Figure 2 presents in more detail the role of the components from the presented digcovery architecture. The green component presents the service layer, whose purpose is mainly to build the interfaces with the client applications and users through Web Services such as RESTFul or through enterprise communications interfaces such as JSON. On top of

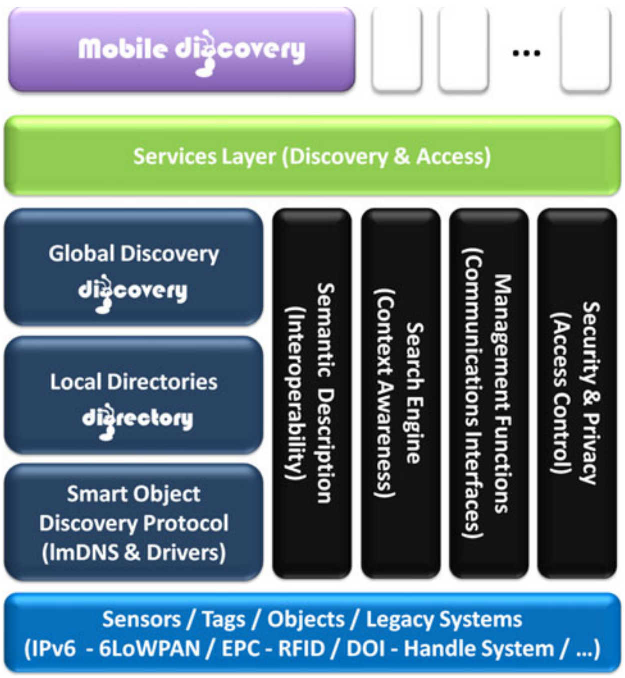

Fig. 2 Digcovery architecture components

this are integrated solutions, such as the application mobile digcovery for the usage of the digcovery architecture through mobile platforms in order to offer a higher interaction with the real world and context-awareness.

The dark blue components present the key components designed, proposed, and developed in order to provide an homogeneous and interoperable environment to discover, lookup, and register services and resources. The main element is the digcovery, which is the global discovery platform. This platform is used to locate the different domains and the wide deployed directories within the different resources. The following elements are the directories, which contain resource and service descriptions from each one of the domains. These directories are not technology dependent; therefore, this will be connected with any other platform through a driver. The platforms and the drivers considered are for platforms such as the electronic product code (EPC) information system for RFID tags, and the handle system for Digital Objects Identifiers (DOI). Finally, a Smart Object Discovery Protocol based on current IPv6-based discovery protocols has also been proposed in order to enable the interaction between IPv6-enabled devices and the directory from its domain. Specifically, a lightweight version of the domain name systems (DNS) extensions for local discovery based on multicast, the called mDNS, and the DNS Service Discovery semantic to describe services and resources over DNS has been defined.

The black components are the other ones key buildings blocks from the digcovery architecture. The first key component is the semantic description; it is a very important issue in order to provide a powerful Open Service Layer. For this purpose, several of the actions have been carried out in EU projects such as SPITFIRE, from the European Commission with the support of events such as the Interoperability PlugFest in conjunction with Probe-IT project, and standardization groups such as IPSO Alliance [8], ETSI, and the most recently released eM2M and the lightweight M2M (LWM2M) from the OMA [10].

The second key component is the search engine; this is the key element of any discovery solution in order to make it powerful. Digcovery has integrated ElasticSearch allowing context-awareness lookup for the digcovery mobile, with some extensions based on geo-location, identification, application profiles, and the support for multiple domains. The third key component relates to the management functions and communication interfaces required to interoperate with third-party platforms and solutions. CoAP has been considered to be compatible with the current Internet of things trends, SenML, and JSON in compliance with the IPSO alliance and the IETF trends for constrained devices. The service discovery offered by the digcovery architecture provides a convenient way for applications to discover and register network services, avoiding the need to manually enter IP addresses, server names, or port numbers. The following section presents the resources integrated into the digcovery architecture.

3 Real-world integration

The architecture presented considers the core, the value chain/network from the end-product to the backend systems from the manufacturer, and the global ecosystem which encloses this.

The Internet of things ecosystem is not only composed of IPv6-enabled devices and tiny objects with communications and processing capabilities, so-called smart objects. This ecosystem is composed of a diverse species of objects, from its origin the things from the physical world can be located.

The sensors and actuators from the physical world are located at fragmented solutions, since different vertical solutions with little overlap can be found. Different vertical solutions satisfying their specific needs such as RFID and NFC for transportation in terms of logistics and ticketing, respectively, can also be found. Health care with continuous patient monitoring through wireless sensor networks based on ZigBee Health Device Profile, 6LoWPAN, and Bluetooth, with specific requirements and needs such as mobility, real time, and reliability. Building and home automation with legacy technologies and proprietary protocols for lighting, heating, cooling, and security which require real time and reliability can also be considered.

Energy for smart meters and smart grid requires mainly reliability. Then, retail with smart tags that are replacing barcodes requires energy-efficiency, low cost, real time, and scalability.

The requirements mentioned need to be met with the characteristics of available technical alternatives and standards from ETSI and OMA for machine-to-machine (M2M) architectures that are focused on cellular networks with support for reliability, low energy-efficiency, high-cost, and relatively bad scalability.

IETF with the Working Groups ROLL, 6LoWPAN [14], CORE Applications [11], Lightweight Implementation Guidelines (LWIG) [15], and COMA are focused on constrained devices and sensors to provide low-cost protocols with high energy-efficiency and scalability, but offer no mobility support. EPCGlobal for RFID with low-cost, real time; wireless identification only, sensing capabilities are not yet standardized.

And finally, handle systems for Digital Object Identifiers (DOI) with low-cost, real-time identification only do not offer a pre-defined medium to carry out identification, usually via a barcode or physical identification. This integration of different resources is handled by digrectory. Digrectory is responsible for collecting either by NFC, 6LoWPAN, WebServices, or mDNS the offered services by various devices that inhabit its domain. Digrectory manages the new registered resources and enables Digcovery to look them up through the search engine presented in the Sect. 5.

Figure 3 presents the resources considered, which are mainly smart objects identified through an IPv6 address. These objects are connected through technologies such as 6LoWPAN, lwIP, IPv6 addressing Proxy and GLoWBAL IPv6 [16]. In addition, other physical resources identified

Fig. 3 Digcovery integration

through the DOI via the handle system [17, 18], or RFID resources identified through EPC and universal identifier digital (UID) [19] have been considered. Finally, other resources can be integrated through Web Services such as bus tracking services [20], weather forecast, and other public Web Services available with relevant resources to build mash ups and mobile platforms.

3.1 Smart objects: IP-enabled devices

The Internet protocol (IP) suite is a suitable solution to realize an Internet of things (IoT), a network of tiny networked-embedded devices that create a link to the physical world. The narrow waist of IP can be used to directly access sensor readings throughout a sustainable city, acquire the necessary information for the smart grid, or control smart homes, buildings, and factories seamlessly from the existing information technology infrastructure. The Internet protocol helps to abstract the complexity from lower layers in multiple aspects such as routing over lossy links, link layer adaption, and low-power communication. These aspects, even when they are complex, are addressed through lightweight implementations that can run on resource-constrained devices such as sensor nodes with only microcontroller units (MCUs), ~10KiB of RAM, and ~100KiB of ROM [15].

This potential for offering Internet capabilities through lightweight implementations is what have during the last years made a higher cyber-physical world integration, reaching the Internet of things through the IP-enabled devices feasible.

The first IP-enabled devices were powered by lwIP and uIP, both stacks provided by SICS [21]. This initial approach was mainly focused on providing a lightweight implementation of the IP stack, lightweight in terms of footprint, i.e., RAM and ROM. Lightweight implementations of stacks were also required. These are not only very important in wireless sensors with constrained resources in terms of memory, but also in terms of energy and communication capabilities (bandwidth). For that reason, new lightweight implementations of IP such as 6LoWPAN [14] and GLoWBAL IPv6 [16] were proposed.

6LoWPAN is a protocol defined by the Internet engineering task force (IETF), which extends wireless sensor networks (WSN) to the Internet and adds an adaptation layer to IEEE 802.15.4 in order to support IPv6. It specifies a wireless link for low-power personal area networks (LoWPANs), such are characterized by more limited capabilities than other WPANs (e.g., Bluetooth) and WLANs (e.g., WiFi). 6LoWPAN has a small frame size, low data rate, limited bandwidth, and power transmitter. In addition to 6LoWPAN, other approaches for the integration of IP in smart objects have been defined in previous works.

Specifically, GLoWBAL IPv6 [16], which offers an aliasing scheme to offer a lightweight integration of IP over layer 4 and layer 5, avoiding the requirements to change layer 3 or integrate adaptation layers such as required by 6LoWPAN. GLoWBAL IPv6 is communication stack independent; this is useful to enable integration with IPv6 for existing networks based on closed stacks over technologies such as the Bluetooth Low Energy and IEEE 802.15.4. GLoWBAL IPv6 allows the integration of IPv6 with more technologies than is feasible merely with uIPv6 and 6LoWPAN.

The devices integrated through IP go from printers, lights, health care systems, and smart grid. The IP-enabled embedded systems and consumer devices already available such as printers, cameras, TVs, and heating control systems, need to be taken into account. Additionally, how to integrate the new generation of IP-enabled smart objects such as the defined over 6LoWPAN and IEEE 802.15.4g will also be considered.

IEEE 802.15.4g protocol extends the IEEE 802.15.4 protocols for its exploration in Advanced Metering Infrastructure (AMI) for Smart Cities deployments. Proof of this new generation of IP-enabled devices has been integrated in the mobile digcovery through a smart lighting solution. Figure 4 presents the smart lights deployed in the street next to the Computer Science Faculty from the University of Murcia. These lights are powered with IEEE 802.15.4g communications and the Applications Profile [9] from the IPSO Alliance [8].

Figure 5 is presented as IP-enabled systems are integrated from Smart City systems as in the case of the street lights to resources from smart buildings such as smart plugs and other sensors from opportunistic devices, e.g., the sensors available in the smartphones.

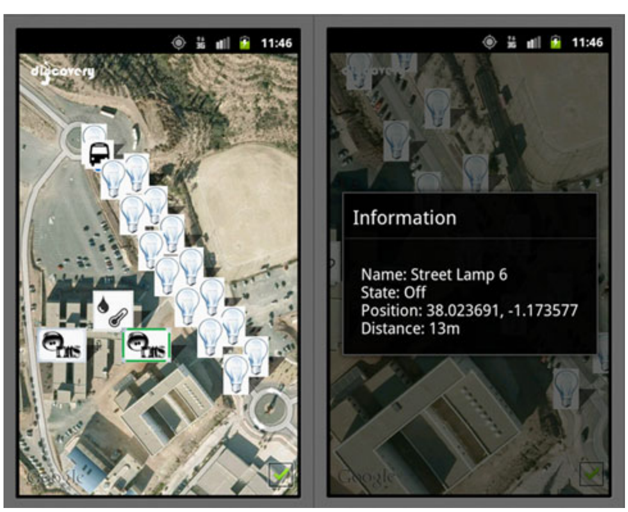

Fig. 4 Smart street lighting enabled with IEEE802.15.4g and IPSO profile

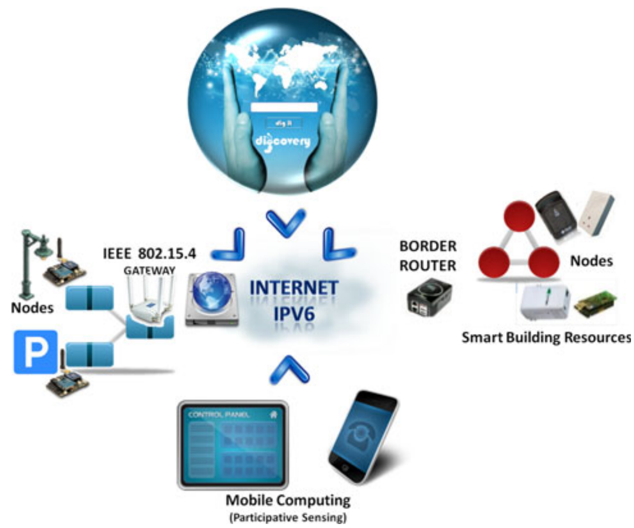

Fig. 5 IP-based integration of resources

3.2 Legacy technologies

The devices with relevant data for the development of mash-up solutions for monitoring and smart metering solutions are not limited to IP-enabled devices, several legacy technologies for gathering the data available from existing sensors in order to reach a higher context-awareness level also need to be integrated.

The architecture mentioned has defined digrectories which act as drivers between native services interfaces and the CoAP-based interface, in order to map to the digcovery interfaces, thereby supporting domains and subnets with different physical layer technologies using CoAP and similar naming conventions.

The digrectories adapt the legacy or proprietary devices from the subnets/domains with different physical layer technologies using CoAP to integrate the different application layer protocols and different naming conventions. For example, CoAP over BACnet, CoAP over Konnex/EIB, or CoAP over X10 [22].

In addition to services and resources adaptation, a mechanism called IPv6 addressing Proxy [23] has been proposed that maps the addressing from any native or legacy addressing or indexing space to IPv6. Therefore, any legacy devices can be addressed through IPv6 addressing globally, unifying the access to any device, resource, or service through IPv6 which avoids using different addressing, indexing, or naming spaces.

Finally, legacy systems have been integrated through its digital identification, so physical objects can be found, which have a RFID tag attached, as a smart thing. Also, several properties and features of these objects are gathered through the extended description offered by the digrectory. In the same way, as is being carried out for RFID through the EPC, Digital Object Identifiers (DOI) through the handle system are considered. By this, means, books, documents, and other virtual objects can be supported. Section 4 presents how to interact with the real world through the identification of devices through identification systems based on EPC and DOI over RFID tags and QR codes.

3.3 Bus tracking through Web Services

Embedded Web Services in the smart objects [13] and the available Web Services from existing resources present another way for integrating relevant data from the real world.

Web Services offer simple and intuitive interfaces to interact with other platforms and systems. RESTFul is the predominant technology to build the Web Services architecture for the IoT, and over the RESTFul architecture, lightweight protocols such as CoAP for embedding Web Services are being defined [13]. For example, digcovery integrates the bus tracking solution offered by the Public Transport Systems Organization from Murcia (Spain) [24]. This Web Service offers a set of resources about bus stops, real-time tracking, and estimated time for arrival for each one of the combinations of bus lines.

A request for the bus tracking integration is specifically based on RESTFul in the Listing 1. This request requires a POST message with a ¡key, value¿ pair. The key is "txtpeticion" and the value the number of the bus stop, in this example "2418", which is the identifier for the bus stop next to the Computer Science Faculty from the University of Murcia. This query can also be for the line number using the prefix "L," e.g., L39 for the bus line from the University to the city center.

Listing 1 Query to the bus tracking Web Service

```
POST servicioweb/response.php HTTP/1.0
Host: www.entidadpublicadeltransporte.es
User-Agent: Mozilla/4.5 [en]
Accept: text/html
Accept-language: en
Accept-Charset: iso-8859-1

txtpeticion=2418
```

Listing 2 Reply from the bus tracking Web Service

```
HTTP/1.1 200 OK
Date: Tue, 18 Sep 2012 11:53:53 GMT
Server: Apache/2.0.40 (Red Hat Linux)
Last-Modified: Tue, 18 Sep 2012 11:53:53 GMT
Accept-Ranges: bytes
Content-Length: 428
Connection: close
<HTML>
...
P2418 FACULTAD BELLAS ARTES
L39A CAMPUS ESPINARDO En parada.(A)
L39C CAMPUS UNIVERSIT. 17 min.(A)
L38A CAMPUS UNIVERSIT. 21 min.(A)
L39C URB. AGRIDULCE II 77 min
...
</HTML>
```

In addition to the Web Services offered by the Public Transport Systems Organization, the functionality is extended with the ElasticSearch engine integrated into the digcovery architecture. This offers new services such as the ability to locate the available bus stops around a position.

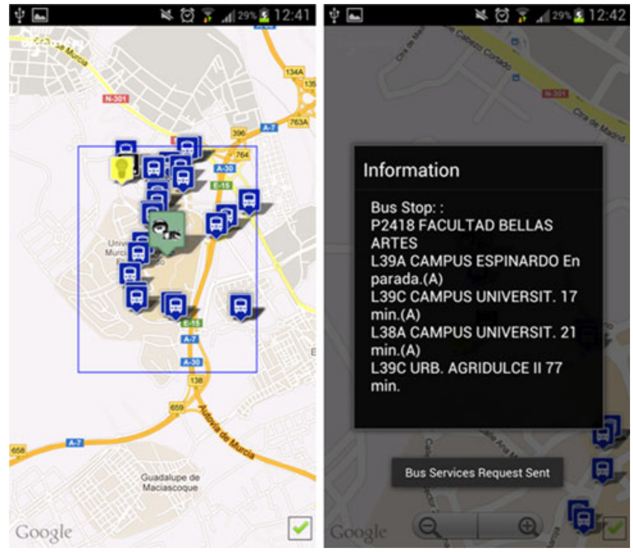

Fig. 6 *Left* bus stops discovery. *Right* reply for a specific bus stop query

Figure 6 presents a screenshot of the application when the bus stops available around a specific position are looked up, specifically the presented picture corresponds to the Campus of the University of Murcia. A bus stop query can be run after in order to get additional details. For example, Fig. 6 presents the next buses for the stop "2418." The reply is presented in the listing 2, of bus lines and the estimated times of arrival.

4 Real-world interaction

In this smartphone era, interaction with the real world is carried out through several technologies integrated in smartphones and mobile platforms. Smartphones are enabled with geo-location capabilities through GPS, real-time location systems. and assisted geo-location through the deployed WiFi networks. Therefore, capabilities to obtain the current location of a smartphone and filter resources and check available services are already available. One of the existing trends is the so-called augmented reality, where the reality is extended with meta-information obtained from the virtual world. In addition to location capabilities, it is feasible to carry out a direct interaction with the objects through its identification. Historically. identification has been carried out with barcodes for global trade identification of the products. Nowadays, a new generation of barcodes called QR codes has been defined, which extends the legacy one-dimension barcode to two dimensions in order to integrate higher quantity of data. Finally, new technologies for identification based on radio

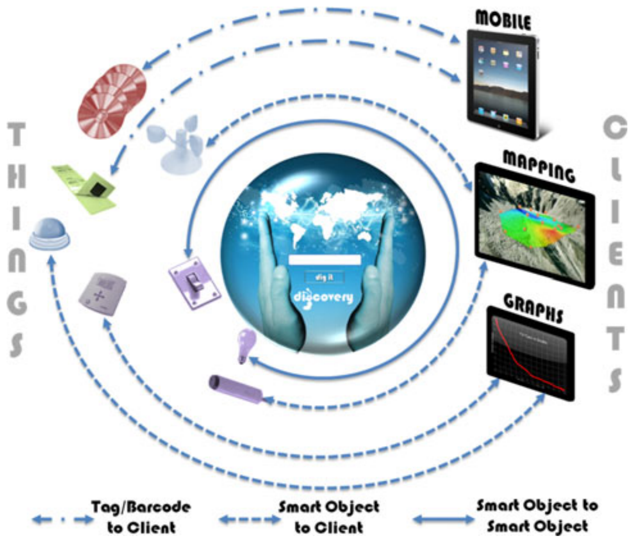

Fig. 7 Interaction between with things, tags, and clients

frequency such as RFID and NFC are making possible the identification of devices and objects without any direct vision higher distance than with the barcodes, obtaining also a high quantity of data, and which allow multiple devices to be read simultaneously.

Smartphones are location enabled, cameras are available to read barcodes/QR codes, and finally the last generation of mobile platforms is powered with NFC that allows interaction with just an approaching between the smartphone and the objects. Figure 7 presents the interactions between clients with the resources available in barcodes/tags and smart objects. The following subsections describe how mobile digcovery uses the existing location and identification technologies in the mobile platforms in order to build a context-awareness solution interaction and exploitation of the resources integrated through the digcovery architecture possible.

4.1 Barcode/quick response codes

Most of the current smartphones are equipped with a camera. Hence, a solution based on this may be used in order to scan barcodes (1 dimension) and QR codes (2 dimensions). Smart things can be identified by reading barcode and QR codes. On the one hand, a European Article Number (EAN-13) can be read, which is mapped in a global level by the GS1-13, EAN-13 which is the main scheme used throughout Europe for retail article numbering. This offers a 12 byte code (since the last is for checksum).

Since the barcode is limited to the article number (12 bytes), QR code which offers higher storage capabilities can also be used. This therefore offers an identification number in the same way as the barcode, but this can also

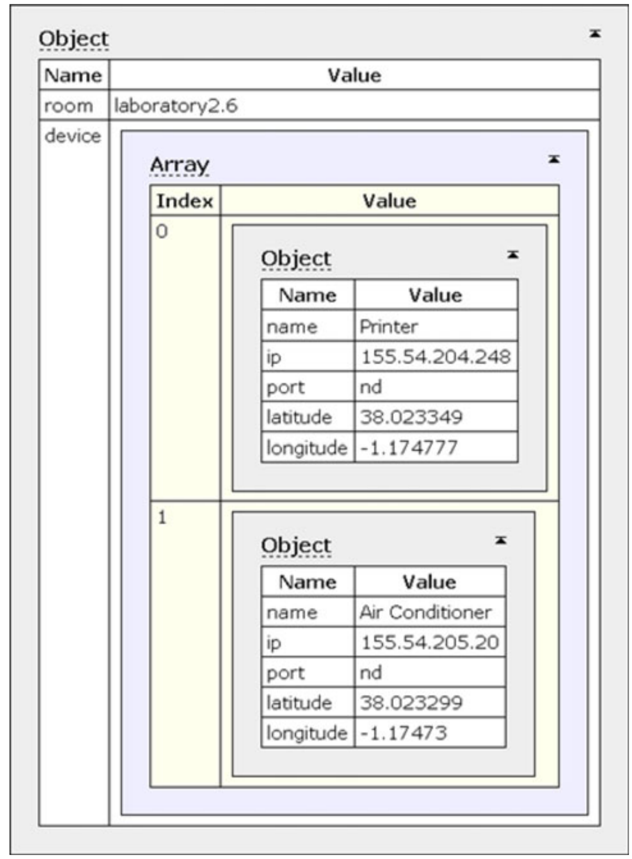

Fig. 8 Printer discovery data structure

link to Internet resources through the inclusion of a universal resource locator (URL), and this can also include descriptions of the product or device.

The description of the product or device is formatted in JSON in order to structure it and make a post-processing and smart search feasible, as described in the Sect. 5. The stored data provides details about the connectivity information such as IP address and port to reach it through the network, and this can also includes additional details such as the location position to display the devices in the map. The listing 3 presents an example of the content for a QR code. That content is formatted in JSON following the format presented in the Fig. 8.

Once the tag is parsed, its details and resources are displayed in the mobile digcovery application to allow the interaction of the user with the devices. The example presented contains the information of a printer and an air conditioner available in one of the laboratories from the Computer Sciences Faculty in the University of Murcia. The Fig. 9 presents an example of the interaction with the QR code, for this project, the smartphone used is a Samsung Galaxy S-III, which is based on Android. Android allows the integration of multi-format 1D/2D barcode decodification through the open source ZXing library [25].

Listing 3 Example of resource in JSON format

```
{
    ``room'': ``lab2.6'',
    ``device'' : [
        {``name'': ``Printer'', ``ip'': ``155.54.204.248'',
            ``port'': ``nd'', ``latitude'': ``38.023349'', ``
            longitude'': ``-1.174777''},
        {``name'': ``Air Conditioner'', ``ip'':
            ``155.54.205.20'', ``port'': ``nd'', ``latitude'':
              ``38.023299'', ``longitude'': ``-1.17473''}
    ]
}
```

Fig. 9 Printer discovery information presented in the digcovery mobile application through the QR interaction

4.2 RFID/NFC

RFID is one of the main technologies for IoT, since the IoT concept was originally conceived as an analogy of the capabilities from RFID to identify any object around the world in a global and unique way. RFID improves the solutions based on barcodes as RFID cards and tags allow greater information storage, while barcodes only allow storing one ID. Extra memory is useful for storing information related to the objects as has been presented for the QR codes.

RFID also offers advantages with respect to the QR codes in terms of usage and dynamic status of the information. First, the interaction with the RFID card/tag is carried out with just approaching, while the barcode and QR code require focus with a camera. Therefore, RFID card/tag usage by elderly people and children is feasible. Firstly, the RFID tag/card can be read from a higher distance than barcodes/QR codes, and multiple tags/cards can be read simultaneously. Secondly, the data stored in a RFID card/tag can be updated [26], while data stored in the barcodes/QR codes are static, since it is defined in the printed code.

RFID brings with it security advantages [27]. This is important since some resources should not be published publicly, as is the case with home automation systems, private devices, and printers. These smart devices should be available for interaction but only by the users permitted, i.e., owner, family, or people inside a company. These devices or services are usually physically accessible only by the owner. The RFID technology is of great help because a RFID tag could contain the information for certain services or devices available and accessible in the physical space such as home automation, security systems, printers, and heating, ventilating and air conditioners (HVAC). In addition, these cards/tags can be protected with a password to avoid invited people reading it and taking control of our systems. A less protected system is the QR code, for example a printer can have a QR code on the side or at the table that indicates the information needed to reach the device and use it to print documents from a smartphone.

With regard to smartphones, the RFID version integrated in smartphones, i.e., NFC, is also gaining high relevance because of its usability and capabilities in terms of availability in the new generation of smartphones. Therefore, NFC/RFID tags and cards are chosen for identification from the new generation of solutions based on IoT for the mobile digcovery.

Mobile digcovery implements the ability to read ISO 14443 and ISO 15693 RFID tags, the 14443 type tag is the common tag used in several environments, some examples of these types of tags are MiFare and DesFire tags. The ISO 15693 is a less common tag; in our case, these tags are used in every room, i.e., classrooms, offices, and laboratories, in the University of Murcia to indicate the patrimonial code. The patrimonial code is used to manage the inventory of a location (room). These tags are available in the wall at the side of the door and they have been extended with the

Fig. 10 Reading the RFID tags available on the walls from the University of Murcia

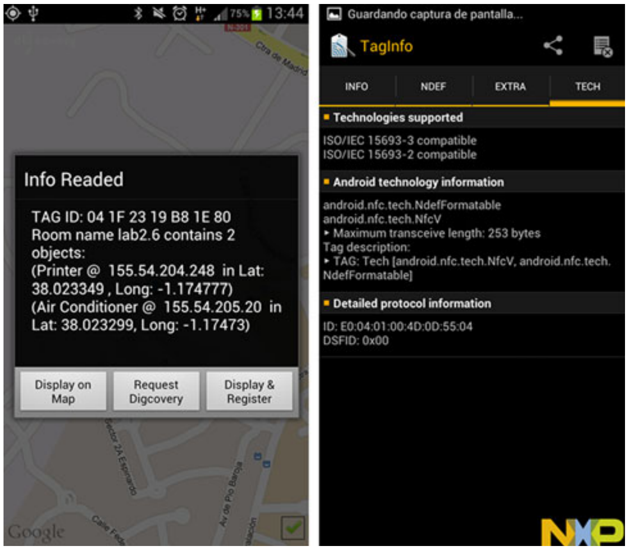

Fig. 11 Printer discovery information presented in the mobile digcovery application through the RFID interaction

information of the available services and devices in the room such as printers and HVAC systems. Figure 10 presents the mentioned patrimonial tags.

The information available in a tag can be easily read, updated, and matched with the corresponding directory disposed to store the information of the devices available in schools, commercial centers, enterprises. More specifically, the UID is used to obtain additional information from the digrectory when the tag has been previously registered.

Figure 11 presents the interaction of the user with the tags and the information gathered from them. It can be seen that information is also formatted in JSON such as in that presented for the QR codes. The format and storage capabilities from RFID and QR code are equivalent, but with the described disadvantages of QR in issues such as that the QR code is static, RFID can store dynamic data, and finally that RFID tags are not increasing the size when the data capacity is increased, i.e., it can be defined nanosize tags with over 1KBits, while 1KBits of data in QR code require a very big size QR code.

In addition to the tag presented, little tags can be integrated to the smart things, for example the Integrated Circuit (IC) model MiFare Ultralight (MF0ICU1) from NXP Semiconductors. This is compatible for RFID and NFC, i.e., ISO/IEC 14443-2 and 14443-3 (Type A) and has a memory size of 512 bits (64 bytes). This reduced size tag is designed to store an identifier, or a URL through technology such as NDEF from NFC. In addition to the tag mentioned, bigger tags can be also considered in particular the I-Code SLI, which presents a doubled memory size, i.e., 1024 bits (128 bytes). An analysis among the different tags and the rest of identification systems is presented in the previous work [28].

4.3 Location

In addition to filtering by resource type, a way to make the discovery for the end user easier, in environments such as Smart Cities, is to discover services that are around the user.

The meaning of close is very different that from the networking and physical point of view since close in networking means under a common domain, over a link-local, which is usually mapped to a specific location. However, when you extend it through Virtual LANs and tunnels, this loses the meaning of close in terms of distance. At the same time, with the proliferation of wireless networks such as 3G, LTE, WiFi, and Wimax, you can be located next to one device, but belong to totally different domains.

For that reason, when addressing a discovery solution for domains such as Smart Cities, a global service discovery needs to be considered. Where the integration of multiple domains is independent of the location, i.e., multiple domains from a similar location can be integrated, then apply close concepts and neighborhood concepts through the interaction with the environment, i.e., physical interaction with tags and QR codes, and mainly with the context-awareness reached by the geo-location of the devices over latitude and longitude coordinates.

A search engine optimized to offer geo-location-based queries and resource types-based lookups is presented in Sect. 5. The integrated search engine allows management of multiple domains with heterogeneous resources and services, even when multiple types of resources and services are stored without a class, resource type, or location organization. As a result, this offers geo-location and context-awareness filtering that has been integrated through the ElasticSearch module from digcovery. The listing 4 presents an example of query for geo-location filtering over the ElasticSearch module.

Listing 4 Example of location filtering

```
'{
``query'' : {
    ``filtered'' : {
        ``query'' : {
            ``range'': { ``longitude'' : { ``from'' :
              ``37.997'', ``to'' : ``37.999'' } }
        },
        ``filter'' : {
    ``range'' : { ``latitude'' : { ``from'' : ``-1.142'',
      ``to'' : ``-1.140'' } }
        }
    }
}
}'
```

5 Search engine

The digcovery search engine is based on ElasticSearch, a document-oriented data base that allows queries to be made through JSON language. The structure of JSON allows for high complex queries to filter and obtain the specific results in a low time.

ElasticSearch has been developed in Java and can be requested from HTTP, i.e., the requests will be sent via Web Services.

ElasticSearch is an interesting tool for storing and retrieving stored data quickly and offers the possibility of getting JSON operations through the integration with digcovery architecture, providing to digcovery a scalable, distributed, and very fast database with multiple access options, as well as a search engine based on RESTFul. ElasticSearch will be used for digcovery in order to, on the one hand, collect unorganized data from the different digrectories and make its lookup and filtering based on services and resource type possible, and on the other hand, offer context-awareness solutions based on geo-location.

Digcovery presents an elastic architecture; this means that several digrectories will be integrated with very different types of resources from different locations. This integration is flexible and elastic, providing a mechanism to carry out global organized lookup.

Current solutions for discovery in the IoT such as the CoAP Discovery, described in the RFC6690 [29], defines the lookup/query based on resource types, i.e., this allows filtering of resources to be discovered specifying a resource type (rt) in the query. Similar types of queries are supported in digcovery through this elastic architecture. The main advantage of ElasticSearch is that it offers the architecture and mechanisms required to manage a distributed and heterogeneous set of repositories (digrectories), in an optimal time, and organizes results filtered by resource type (e.g., light). CoAP discovery on the other hand is limited to retrieving the results from the local domain or from a single repository. Therefore, ElasticSearch offers the same potential of CoAP Discovery without the limitations of discovery in local domain (multicast-based) or to the resources of a centralized server (CoAP Resource Directory).

5.1 Digcovery database structure

As we have mentioned before, the request will be made via Web Services with an HTTP request. An HTTP request consists of the following format described in 5.

Listing 5 Example of the URL structure for the Search Engine Query

```
URL:9300/[index/subnode1/subnode2/subnodeN/]_search?q=    {
    (JSON query) }
```

Index and subnodes are not necessary, but it is useful to insert data in a specific subnode to have the data ordered and to filter the queries requests more efficiently.

In our case, the index is the domain name in order to separate the services allocated to each one, and the subnode is similar to a DNS pointer that means the group of services allocated in the domain.

5.1.1 Subnodes

The implemented database offers two ways to request the information. Firstly, it is possible to access all data (domain, services, pointers, txt) in a single access in order to make general queries over a domain, and secondly it is possible to access the data from a specific service. These types of subnodes are the digcovery generic pointers, "any" to access a domain data and "services" to access services data.

This offers the possibility to define new types of pointers specifically for a certain domain which represent multiple resources. For example, this allows "Lights" to be defined as a representation of the group for all the lights available in that domain, where the domain can be represented at physical level by street, building, floor, or campus level.

5.2 Communication with digcovery search engine

5.2.1 Request for the mobile digcovery

Communication with the digcovery search engine is based on JSON. Therefore, the query is modeled by an object that specifies the query code and the associated parameter object. The query object for the generic digcovery request structure is presented in the listing 6.

Listing 6 Digcovery Generic Request Structure (GDRS)

```
Object QueryRequest{

        code: ''code value'',
        data: ''parameter object''

}
```

This object is converted to JSON and sent via UDP. There are several command codes to make queries to the digcovery server. For example, the "any" code allows clients to request all domain data that includes services, pointers, and txt.

The "geo" code, allows the mobile digcovery to obtain the position of surrounding services. The parameters object is different for each type of code request; in this case, it is interesting to explain the "geo" parameter object being used by mobile digcovery. The parameter object for "geo" code is presented in the listing 7.

Listing 7 Example of GeoPoint Object fit in digcovery query structure

```
Object GeoPoint {
        latitude: x,
        longitude: y
        distance: m (meters)
}
```

The digcovery server receives this object and parses it to make the necessary query to filter the results returned by ElasticSearch and send the responses to mobile digcovery. This parser calculates the quarter allocated between our position and the max distance of the query. This allows a located search to be to carried out in order to enhance search efficiency. This means that a mobile digcovery request is parsed as a ElasticSearch query, such as is presented in the listing 8.

Listing 8 Example of GeoLocation ElasticSearch Query

```
  query   : {
          filtered   :{
                  query   :{
                          range   :{  longitude  :{
                            from   :   37.997   ,   to
                          :  37.999   }}
                  },
                   filter   :{
                          range   :{  latitude  :{
                            from   :   −1.142   ,   to
                          :   −1.140   }}
                  }
          }
}
}
```

5.2.2 *Response for the mobile digcovery*

The Digcovery response has several formats depending on the code request. In the “geo” code case, the response will be the called ServiceJS array. A ServiceJS is a compressed representation of a Service from Database. This object contains the fields presented in the listing 9.

Listing 9 ServiceJS-based representation

```
Object ServiceJS{
        String name,
        Int port,
        String addr,
        String gps,
        String serverAddr,
        Double latitude,
        Double longitude
}
```

This array is received in JSON format with the necessary information about a service. The name of the service has the DNS-SD format; therefore, it is possible to communicate with the sensor (if is a sensor) with the protocol type included in the name.

For example, potentially a CoAP packet can be sent to obtain the temperature value from the resource named “temperature._coap._udp.” The address of a ServiceJS contains the IPv6 address of the sensor service; this means that to connect via IPv4, the serverAddr (address of the server) should be used. Finally, the server offers the DigcoveryRedirector module, which translates from IPv4 to IPv6 in order to reach the sensor in the 6LoWPAN, GLoWBAL IPv6, or uIPv6 network, such as described in the Sect. 3.1.

5.3 Digcovery CoAP service protocol (DCSP)

A CoAP interface is offered to communicate with any service or resource provided through CoAP. A simple application protocol to get specific commands, defined by each sensor disposes, over the ElasticSearch engine.

This interface allows checking of the command list, such as the Get command defined in HTTP, and reused for Web Services over the REST architecture. This command is necessary to discover the specific command for each sensor so it can communicate via CoAP. The listing 10 presents the structure for the CoAP protocol.

Listing 10 Get command

```
coap://GET cl
```

That command returns an array of command interface in JSON format as will be shown in the listing 11. This output can be returned by a device composed of two devices, a light actuator and a temperature sensor. This presents the GET, PUT, and DEL available commands. For example for the temperature sensor, GET commands are only available, since it is a sensor and offers read-only resources. Another example is the light actuator, which also offers PUT commands since it is an actuator and offers resources to act over the device, for example in this case switching the light.

Listing 11 Example of command interface sent by the sensor as response to the previous get command

```
[
        {
                ''name'':''_coap._udp.light''
                ''GET'':''light_status'',
                ''PUT'':''switch'',
                ''DEL'':''null''
        },
        {
        ''name'':''_coap._udp.temperature''
        ''GET'':''temp_status'',
        ''PUT'':''null'',
        ''DEL'':''null''
        }]
```

Regarding the JSON structure, the name field is the command type that must be sent via CoAP (except “name” field) and the value is the particular command of the sensor to interact with it. For example, listing 12 presents a get command for the temperature of the sensor with the “temp_status” payload.

Listing 12 Get command for specific resource

```
coap://GET temp_status
```

Otherwise, if the light is required to be turned off, the packet presented in the listing 13 is sent, which puts the value 0 over the resource switch.

Listing 13 Put command to establish the status for a resource

```
coap://PUT switch

payload: 0
```

6 Related works

As has been presented, the IoT provides connectivity between people, devices, services, and smart objects allowing the development of new ways of interacting with the environment.

Several works have been proposed for interacting with the Internet of things. Many of these works are related to smartphones, since these consumer devices offer a huge set of capabilities to process and connect with the things that are located surround the users.

Some works are related to ways to connect with sensors. Many others relate to how to carry out interaction between the applications and the sensors/services among the works related to the latter are the ways to connect with wireless sensors. Some of the works closer to those presented in this contribution address the connectivity of smartphones with sensors connected to wireless sensor networks, and many others that can be reached through the Internet infrastructure through Web Services.

Specifically, there are several ways to reach a sensor located in a sensor network from a smartphone, the first way is to adapt the 802.15.4 hardware to the smartphone. The work located in [30] presents uSD hardware that uses a SD card port to bring ZigBee connectivity and a middleware to manage the hardware and interact with the sensors. Another example of adapted ZigBee hardware is located in [31], where a custom hardware connected via USB reaches clinical devices and allows the smartphone to act as a gateway to connect the devices mentioned with the backend. Other ways to reach a sensor that is within a wireless sensor network is to use a gateway that offers Web Services. Similarly, the work defined in [32] presents a middleware to carry out the access to sensors and cameras from the user.

The majority of the works available in the state of the art for the Internet of things connectivity for smart objects present IPv6 and 6LoWPAN as the most suitable protocols to reach global connectivity for a sensor. The work carried out in [33] presents an example of how to reach a temperature sensor through a smartphone enhanced with 802.15.4 physical medium using 6LoWPAN network protocol to access the sensor.

Some works and applications use the smartphones to exploit the services offered by sensors. These works present how to use a backend to obtain information or even where the smartphone is used as a sensor or bridge to communicate with the backend. These applications are related to several environments or topics. The work presented in [34] uses a smartphone with iOS to access a WebServer that manages a home automation security system based on X10. An example of the capabilities from that work in terms of machine-to-machine communication is that when a motion sensor detects any activity, this activates the cameras to record the environment and advise the homeowner. The work presented in [35] implements a framework for body area sensor networks based on Bluetooth where they use the smartphone as a gateway to transport the information gathered from the body sensors and processed into the smartphone to the backend.

There are other many smartphone applications applied to transport, e.g., the work presented in [36] enables the remote control of a car through OSGi-based middleware and the communication of control commands within the car is carried out through control rea network. Another application that relates to the smartphone with sensors in transport is located at [37]. In this case, the spaces available in a car park is detected through a wireless sensor node that uses a luxometer to detect the presence of a car in the parking lot. The backend manages the use of the parking and reservation service, which can be reached through a cellular network or the home gateway to reach the Internet. To conclude the examples of public transport solutions, the document [38] introduces the architecture and application used by Edinburgh city council to carry out a bus Web Service, one of the most popular systems that used currently.

Most applications are accessing wireless sensors integrated in homes, cars, and clinical environments, and Web Services from government or private enterprises that offer several Web Services in a Smart City. This is demonstrated

Table 1 State of the art analysis

Feature/solution	Mobile digcovery	[30]	[31]	[32]	[33]	[34]	[35]	[36]	[37]	[38]	[39]
Backend systems required	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Support for legacy sensors (e.g., X10)	Yes	No	No	No	No	No	No	No	No	No	No
Support for IPv6 sensors	Yes	No	No	Yes	Yes*	No	No	Yes*	No	No	No
Support for web services (e.g., CoAP/HTTP)	CoAP & HTTP	No	HTTP	HTTP	HTTP	No	No	No	No	HTTP	HTTP
Discover resources via IP protocols (e.g., mDNS)	Yes	Local	No	No	No	Yes	Yes	No	No	Yes	Yes
Discover resources via RFID/QR/Barcode	Yes	No	No	No	No	No	No	No	No	No	No
Hardware required (e.g., Zigbee adapter)	No	Yes	No	No	No	Yes	No	No	No	No	No

Asterisks indicates that it is not originally supported in the description, but the technology solution provided and used is able to support it in the case [39], where a smartphone application provides a middleware for U-city, based on cloud computing, to thus enable users to use, manage, and monitor mobile cloud computing for U-city services without expert knowledge. Table 1 presents the comparability among the mentioned works in terms of connectivity, i.e., IPv6 connectivity, interoperability through Web Services, support for legacy technologies, and the required hardware to reach a suitable integration. This also addresses other issues regarding interaction, i.e., support for interaction through RFID/NFC and barcode/QR code. The discussion located in Sect. 7 presents the main differences between the related works and the proposed mobile digcovery.

7 Discussion

The previous section has presented several isolated solutions for bus tracking, middleware for home automation, and finally Web Services-based platforms to integrate sensors with smartphones.

Firstly, in most cases, it is necessary to know the existence of the sensors or services beforehand. That is why the work carried out in this paper introduces an innovation in the sense that this application allows discovery of sensors and services that surround us through geo-location and context-awareness capabilities. These services can be discovered in different ways, either through the backend deployed where anyone can report their services offered to the public or through RFID tags/QR codes to discover services and devices that should only be accessible if given the necessary physical access. After the service or sensor is discovered in our application, it is possible to address the use of this service/device to another mobile application designed for the service. Therefore, mobile digcovery is offering several ways to discover resources, in terms of services and services, through innovative technologies such as RFID/NFC and QR codes but it is not limited to the discovery of the services and devices.

Secondly, the capabilities to integrate and support legacy technologies have been offered. Specifically, digrectories have been defined which act as drivers between native services interfaces and a CoAP-based interface, in order to map to the digcovery interfaces, thus supporting domains and subnets with different physical layer technologies using CoAP and similar naming conventions.

The digrectories adapt the legacy or proprietary devices from the subnets/domains with different physical layer technologies using CoAP to integrate the different application layer protocols and different naming conventions, for example, CoAP over BACnet, CoAP over Konnex/EIB, or CoAP over X10. The IoT6 European Project is currently working in this integration from several building automation technologies to IoT-enabled interfaces (i.e., CoAP and IPv6). Thirdly, in addition to the integration of a legacy device, electronic product code information system (EPCIS) has been integrated; thus, its physical objects can be found, which have a RFID/NFC tag attached, as a smart thing. This offers the benefits that several properties and features of these objects are gathered through the extended description offered by the EPCIS. In the same way, as is being carried out for RFID through the EPC, the DOI through the handle system [17, 18] are also being considered for ongoing work. In this manner, it can also be used for books, documents, movies (DVDs, BRs), and music (CDs, MP3).

Fourthly, mobile digcovery exploits the capabilities to obtain geo-location information by the smartphones and mobile platforms through systems such as real-time location systems based on WiFi networks and the GPS system based on Satellite, in order to reach a higher contextual awareness. For this purpose, this work has presented in a novel way the exploitation of the ElasticSearch architecture for the integration and filtering from JSON-structured data extracted from the resources itself, in order to reach a suitable solution for filtering by resource types and location.

Finally, digcovery can be seen by other platforms as a middleware; for example. it is being integrated with global sensor network (GSN) architecture [40, 41], which is one of the most extended low-power-embedded wireless network middleware architectures. Digcovery allows the export of devices and services discovered directly to GSN framework. It therefore makes the use of management of the services and devices located around the world for personal or industrial purpose simpler. In addition, GSN offers the specific services application to be built over digcovery architecture and allows data collection from the sensors, managing statistics, raising alarms, etc. Therefore, with the options and interfaces offered by digcovery to enable the integration with third-party platforms such as GSN, a flexible and scalable environment is offered to discovery, lookup, manages, and uses the data from the Smart Things, EPCIS, Handle System (DOI), and legacy technologies such as BACNET, KNX, and X10.

8 Conclusions and ongoing works

The quantity of devices and data available is continually increasing. It is estimated that there will be more than 50 billion devices by 2020. Currently, in 2012, we have 12.5 billion connected devices, and since 2008, we have more connected devices to Internet than people in the world reaching a real Internet of things. At the same time, during the last decade, smartphones are presented as a new way to interact with the world and to access the Internet. Now, new mechanisms are required to exploit the available resources through smartphones. Since the number of resources, i.e., services and devices are increasing, higher context-awareness and location-enabled solutions are required in order to filter and make these resources user-friendly for citizens. Therefore, discovery solutions are required. This paper has presented the possibility of utilizing new technologies to enable smart objects discoverable, accessible, available, usable, and interoperable. For this purpose, the digcovery architecture developed in the IoT6 European Project [42] has been defined. Digcovery architecture offers a homogeneous and suitable mechanism for the global resource discovery, devices access for the deployed smart objects in the different scenarios, and the integration of legacy and already available sensors in smart buildings and in Smart Cities. These are required features for a solution based on the Internet of things.

This work has presented integration of different resources available in the real world. Specifically, IP-enabled devices are integrated through novel protocols such as 6LoWPAN [14] and the developed protocol GLoWBAL IPv6 [16], Web Services-based resources such as the bus tracking solution from the Murcia Region, and finally legacy resources from building automation and identification systems such as those based on DOI and electronic product codes.

The integration of the mentioned subsystems is part of the IoT6 work, a collaboration of several institutions formed for this specific purpose, for example, the Technical University of Vienna in the integration of Building Automation subsystems [22], and the Korea Advanced Institute of Science and Technology in the integration of the EPCIS [19].

This work has presented mobile digcovery. Thanks to the integration reached by the digcovery architecture; it is feasible to exploit the geo-location and context-awareness capabilities from the mobile platforms such as smartphones. Specifically, the interaction through QR codes, RFID tags, and location-based filtering has been presented. Mobile digcovery therefore offers context-awareness capabilities for the discovery of resources via the identification and location subsystems available in the mobile platforms.

The location-based, resource types, and identification filtering have been achieved through a scalable lookup based on ElasticSearch, JSON description of resources, and the discovery of heterogeneous Internet of things resources through the development of the digcovery architecture based on REST. Digcovery offers a high scalability for the discovery based on an elastic architecture, where several resources can be integrated in a distributed way through the different digrectories. Even when a centralized digcovery is defined, its scalability is powered by the elasticity offered by ElasticSearch engine to integrate distributed digrectories.

Ongoing work is focused, on the one hand, on the integration of different legacy technologies and resources in the context of the IoT6 European Project, and on the other hand, to make open and accessible the digcovery platform to promote the integration of sensors and devices from the end users (participative sensing). Regarding this last point, digcovery architecture offers the framework to allow the users to register/include their own sensors into a common infrastructure and access/discover the available resources. In so doing, this will also enable the integration of opportunistic resources and third-party resources from proactive citizens. We therefore consider that digcovery could offer an interesting framework for the integration of resources and interaction of the users with the available services and devices in environments such as Smart Cities, where for example users can lookup bus stops, available printers, climate systems, and any other available or published resources such as parking, weather, and traffic information. There exists a great number of opportunities in domain-specific resources such as defined in shopping centers for products, or defined for health care such as product information for specific illnesses, e.g., diabetes [43], or advice to reduce adverse drug reactions and allergies [28].

Acknowledgments This work has been carried out by the excellence research group "Intelligent Systems and Telematics" granted from the Foundation Seneca (04552/GERM/06). The authors would like to thank the Spanish Ministry of Science and Education with the FPU program grant (AP2009-3981), the Ministry of Science and Innovation, through the Walkie-Talkie project (TIN2011- 27543-C03-02), the STREP European Projects "Universal Integration of the Internet of things through an IPv6-based service-oriented architecture enabling heterogeneous components interoperability (IoT6)" from the FP7 with the grant agreement no: 288445, "IPv6 ITS Station Stack (ITSSv6)" from the FP7 with the grant agreement no: 210519, and the GEN6 EU Project.

References

1. Steven Conner W, Krishnamurthy L, Want R (2001) Making everyday life easier using dense sensor networks. In: UbiComp 2001, proceedings of the 3rd international conference on ubiquitous computing, pp 49–55. Springer, London. ISBN:3-540-42614-0
2. Siewiorek D (2012) Generation SmartPhone, IEEE Spectrum, http://spectrum.ieee.org/consumer-electronics/gadgets/generation-smartphone/, September
3. Saha D, Mukherjee A (2003) Pervasive computing: a paradigm for the 21st century. Computer 36(3):25–31. doi:10.1109/MC.2003.1185214
4. Atzori L, Iera A, Morabito G (2010) The internet of things: a survey. Comput Netw 54(15):2787–2805
5. Emerson B (2010) M2M: the internet of 50 billion devices, Win-Win, Editorial, Huawei
6. Edwards WK (2006) Discovery systems in ubiquitous computing. Pervasive Comput IEEE 5(2):70–77. doi:10.1109/MPRV.2006.28
7. Hoang DD, Paik H-Y, Benatallah B (2010) An analysis of spreadsheet-based services mashup. In: Shen HT, Bouguettaya A (eds) Proceedings of the twenty-first Australasian conference on database technologies—vol 104 (ADC '10). Australian Computer Society, Inc., Darlinghurst, pp 141–150

8. Dunkels A, Vasseur J (2008) IP for smart objects, Internet Protocol for Smart Objects (IPSO) Alliance, White Paper, N. 1, IPSO Alliance
9. Shelby Z, Chauvenet C (2012) The IPSO application framework, draft-ipso-app-framework-04, IPSO Alliance, Interop Committee
10. Tian L (2012) Lightweight M2M (OMA LWM2M), OMA device management working group (OMA DM WG), Open Mobile Alliance (OMA)
11. Shelby Z, Hartke K, Bormann C, Frank B (2012) Constrained application protocol (CoAP), constrained resources (CoRE) working group, internet engineering task force (IETF), work in progress, draft-ietf-core-coap-13
12. Li ST, Hoebeke J, Jara AJ (2012) Conditional observe in CoAP, constrained resources (CoRE) working group, internet engineering task force (IETF), work in progress, draft-li-core-conditional-observe-03
13. Shelby Z (2010) Embedded web services. Wirel Commun IEEE 17(6):52–57. doi:10.1109/MWC.2010.5675778
14. Hui J, Thubert P (2011) Compression format for IPv6 datagrams over IEEE 802.15.4-Based Networks, RFC 6282, Internet Engineering Task Force (IETF), ISSN: 2070-1721
15. Bormann C (2012) Guidance for light-weight implementations of the internet protocol suite, lightweight implementation guidelines (LWIG) working group, internet engineering task force (IETF), work in progress, draft-ietf-lwig-guidance-02
16. Jara AJ, Zamora MA, Skarmeta A (2012) GLoWBAL IP: an adaptive and transparent IPv6 integration in the internet of things, mobile information systems. IOS Press. ISSN:1574-017x. http://iospress.metapress.com/content/x611r3t20n171102/
17. Sun S, Lannom L, Boesch B (2003) RFC3650—handle system overview. IETF Standards
18. Coetzee L, Butgereit L, Smith A (2012) Handle system integration as an enabler in an internet of things smart environment. CSIR Meraka Institute, South Africa, p 111
19. Sung J, Kim D (2009) Sensor profile requirements for sensor network capability information in the EPCglobal network, AUTO-ID Labs, White Paper WP-SOFTWARE-027
20. Biagioni J, Gerlich T, Merrifield T, Eriksson J (2011) EasyTracker automatic transit tracking, mapping, and arrival time prediction using smartphones. In: SenSys, ACM, pp 68–81
21. Dunkels A (2006) uIP-A free small TCP/IP stack, The uIP 1.0 reference manual, Swedish Institute of Computer Science—SICS, 10.1.1.154.2510
22. Jung M, Reinisch C, Kastner W (2012) Integrating building automation systems and IPv6 in the internet of things, sixth international conference on innovative mobile and internet services in ubiquitous computing (IMIS), pp 683–688. doi:10.1109/IMIS.2012.134
23. Jara AJ, Moreno P, Skarmeta AFG, Varakliotisy S, Kirstein P (2012) IPv6 addressing proxy: mapping native addressing from legacy communication technologies and protocols to IPv6 and the internet of things, international workshop on identification, information and knowledge on the internet of things, IEEE international conference on the internet of things (IoT 2012), Wuxi, China
24. Entidad Pblica del Transporte de Murcia (2012) Real time bus tracking through web services. http://www.entidadpublicadeltransporte.com/servicioweb/, Last time accessed 15th September 2012
25. Mackintosh A, Martin A, Brown B, Brunschen C, Daniel S (2012) Zxing, open source library to read 1D/2D barcodes. http://code.google.com/p/zxing/
26. Jara AJ, Alcolea AF, Zamora MA, Skarmeta AFG (2010) Analysis of different techniques to define metadata structure in NFC/RFID cards to reduce access latency, optimize capacity, and guarantee integrity, intelligent manufacturing systems, vol 10, Part. 1, pp 192–197. doi:10.3182/20100701-2-PT-4011.00034
27. Jara AJ, Zamora MA, Skarmeta AFG (2009) NFC/RFID applications in medicine: security challenges and solutions, 5th international conference on intelligent environments, ambient intelligence and smart environments, vol 4. IOS Press, pp 879–884. 978-1-60750-056-8
28. Jara AJ, Zamora MA, Skarmeta AF (2012) Drug identification and interaction checker based on IoT to minimise adverse drug reactions and improve drug compliance. Pervasive Ubiquitous Comput. doi:10.1007/s00779-012-0622-2
29. Shelby Z (2012) RFC6690—constrained RESTful environments (CoRE) link format, IETF standards, CoRE working group
30. Zhang J, Chen C, Ma J, He N, Ren Y (2011) uSink: smartphone-based mobile sink for wireless sensor networks, 5th IEEE workshop on personalized networks (PerNets 2011), pp 90–95
31. Chu Y-H, Hsieh Y-C, Wang C-H, Pan Y-C, Chang R-I (2011) UPHSM—ubiquitous personal health surveillance and management system via WSN agent on open source smartphone, IEEE 13th international conference on e-health networking, applications and services, pp 60–63
32. Moreira N, Venda M, Silva C, Marcelino L, Pereira A (2011) @sensor—mobile application to monitor a WSN, 6th Iberian conference on information systems and technologies (CISTI), p 16
33. Schleiss P, Torring N, Mikkelsen SA, Jacobsen RH (2012) Interconnecting IPv6 wireless sensors with an Android smartphone in the future internet, 2nd Baltic Congress on future internet communications, pp 14–18
34. Das SR, Chita S, Peterson N, Shirazi BA, Bhadkambar M (2011) Home automation and security for mobile devices, 1st IEEE PerCom workshop on pervasive communities and service clouds, pp 141–146
35. Wang S, Shi W, Arnetz B, Wiholm C (2010) SPARTAN—a framework for smart phone assisted real time health care network design, 6th international conference on collaborative computing: networking, applications and worksharing (CollaborateCom), pp 1–10
36. Lee B-H, An S-H, Shin D-R (2011) A remote control service for OSGi-based unmanned vehicle using smartphone in ubiquitous environment, 3th international conference on computational intelligence, communication systems and networks, pp 158–163
37. Srikanth SV, Pramod PJ, Dileep KP, Tapas S, Patil MU, Sarat CBN (2009) Design and implementation of a prototype smart PARKing (SPARK) system using wireless sensor networks, international conference on advanced information networking and applications workshops, pp 401–406
38. Smart Cities Project (2011) Mobile App for bus users in Edimburgh, SmartCities Brief, No. 19. ISBN. 9781907576362. http://www.smartcities.info
39. Park JW, Yun CH, Rho SW, Jung HS (2011) Mobile cloud web-service for U-City, 9th IEEE international conference on dependable, autonomic and secure computing, pp 1061–1065
40. Aberer K, Hauswirth M, Salehi A (2006) A middleware for fast and flexible sensor network deployment, 32nd international conference on Very large data bases (VLDB '06), VLDB Endowment, pp 1199–1202
41. Salehi A, Riahi M, Michel S, Aberer K (2009) GSN, middleware for stream world, 10th international conference on mobile data management (MDM 2009), May 18–20, Taipei, Taiwan
42. Jara A, Skarmeta AF (2012) Deliverable 3.1—look-up/discovery, context-awareness, and resource/services directory, universal integration of the internet of things through an IPv6-based service oriented architecture enabling heterogeneous components interoperability (IoT6), FP7 ICT—288445. http://www.iot6.eu/images/stories/deliverables/IoT6-D3.1.pdf
43. Jara AJ, Zamora MA, Skarmeta AFG (2011) An internet of thingsbased personal device for diabetes therapy management in ambient assisted living (AAL). Personal Ubiquitous Comput 15(4):431–440. doi:10.1007/s00779-010-0353-1