An information-centric and REST-based approach for EPC Information Services

Federica Paganelli, Stefano Turchi, Lorenzo Bianchi, Lucia Ciofi, Maria Chiara Pettenati, Franco Pirri, and Dino Giuli

***Abstract*: Radio Frequency Identification (RFID) techniques are considered relevant building blocks for the Internet of Things. The interoperability across different RFID software and hardware infrastructures is a key requirement for achieving effective and wide-scale Internet of Thing deployments. In this context, the EPC Information Service (EPCIS) is a set of standard specifications for sharing RFID-related data (i.e., EPC events) both within and across enterprises. Although the EPCIS specifies a set of HTTP and Web Service interfaces for querying and adding EPC events, interoperability and easiness of use is hindered by the fact that client applications should be aware of the repositories that are authoritative for one or more given queries and links among related events are not explicitly represented in response messages. In this paper we argue that, by leveraging emerging REST and Linked Data paradigms, EPC events can be handled as a graph of globally-addressable information resources that can be navigated, queried, and aggregated through a uniform interface and seamlessly across organization domains. To validate this approach, we have developed a prototype that exposes the EPCIS interfaces as a set of REST APIs. The prototype implementation exploits the information modeling and management capabilities provided by a framework, called InterDataNet (IDN), that we conceived and developed to ease the realization of the Web of Data and Linked Data applications.**

***Index Terms*— Web Services, Representational State Transfer, RFID, EPC, Linked Data, HTTP, Web of Data.**

I. INTRODUCTION

The Internet of Things vision implies the creation of large-scale networks of “smart things” made possible by of RFID and wireless sensor and actuator networks nodes distributed in the physical environment [1], [2]. According to this vision, *data of real world objects and events will be available globally and in vast amounts. These data will be stored in widely distributed, heterogeneous information systems, and will also be in high demand by business and end user applications.”* [3]. Therefore, mechanisms allowing to easily access, retrieve and manage information resources, while guaranteeing scalability and interoperability, are required [4].

Standards play a key role in achieving these objectives. As far as RFID technologies are concerned, the EPCglobal Network specifies an architecture of hardware and software components and interfaces to efficiently handle logistic processes [5]. The Electronic Product Code (EPC) is a universal identifier used for physical objects. It can take the form of a Uniform Resource Identifier (URI), thus enabling information systems to refer to physical objects. The EPCIS [6] specifies a distributed system of information repositories that store RFID-related events and data. These repositories can be accessed through an HTTP-based Capture Interface for adding new events and through a Query interface based on Web Services (WS) [7] for querying and accessing stored events.

WS specifications represent state-of-the art solutions for enabling data exchange across technological and organizational boundaries in heavy and complex business applications [7]. Nonetheless, we argue that, by leveraging emerging Web Science paradigms [8], EPC events can be handled as a graph of globally-addressable information resources that can be navigated, queried, and aggregated through a uniform interface and seamlessly across organization domains. This information-centric approach help in explicitly model relations among EPC events and data and relies on widely adopted Web standards, thus improving interoperability and data exchange both within and across enterprises.

Our approach relies on the Representation State Transfer (REST) architectural style and Linked Data, which are two paradigms characterizing the emerging interdisciplinary field of Web Science. The REST architectural style was proposed by Roy Fielding in his doctoral dissertation [9] as an architectural style for building large-scale distributed hypermedia systems. The term “Linked Data” refers to a set of best practices for publishing and connecting structured data on the Web [10] via standard technologies (e.g. HTTP URIs [11], RDF [12], SPARQL [13]). Thanks to links connecting data

Manuscript received December 15, 2012; revised February 22, 2013.

The present work is an extended version of the paper entitled “Designing EPCIS through Linked Data and REST principles” presented at the SoftCOM2012 Conference.

F. Paganelli is with the National Interuniversity the National Interuniversity Consortium for Telecommunications, Italy (phone: +39 055 4796382; fax: +39 055 4796427; e-mail: federica.paganelli@unifi.it.

S. Turchi, L. Bianchi, L. Ciofi, F. Pirri, D. Giuli are with the Department of Electronics and Telecommunications, Università degli Studi di Firenze, 50139, Italy (e-mail: stefano.turchi, lorenzo.bianchi, lucia.ciofi, franco.pirri, dino.giuli@unifi.it).

M.C. Pettenati is with the ICON Foundation, Sesto Fiorentino, Italy (e-mail: mc.pettenati@iconfoundation.net)

from different sources, the Web is expected to evolve towards a global graph of data that can be navigated and queried.

By applying this information-centric approach, in this paper we propose: i) a graph-based information model for EPC events and data, and ii) a set of RESTful EPCIS APIs for accessing, navigating and modifying this web of EPC information resources. We thus provide developers with tools easing the building of lightweight Web-based applications for EPC-related data exchange and consumption.

To accomplish these objectives, we have exploited a framework, called InterDataNet (IDN) [14], that we conceived and implemented to ease the development of web-based applications by exploiting REST and Linked Data principles.

The remainder of this paper is organized as follows. Section II gives background information about the EPC Information Services (EPCIS) specifications and Section III introduces the main principles of REST and Linked Data paradigms. In Section IV we discuss related work and motivate our contribution. In Section V we describe the information modeling and management capabilities provided by the InterDataNet framework. In Section VI we describe our information-centric approach for exposing EPCIS capture and querying capabilities and we explain how we have exploited the InterDataNet features to this purpose. In Section VII, we present a case study and finally, Section VIII concludes the paper with insight for future work.

This work extends a previous work [15] by analyzing in more detail the topic and providing more advanced solutions: each event type defined in the EPCIS specifications have been analyzed and modeled in compliance with the IDN-IM formalism (see section VI for details), the IDN Search Service (see section V.C) features have been fully exploited to improve the Query Interface capabilities, and a complete implementation of EPCIS Capture and Query Interfaces enforcing the principles discussed in this paper has been deployed and tested (see section VII).

II. EPC Information Services

The EPCglobal Network specifies an architecture of hardware and software components and interfaces to efficiently handle logistic processes. Its main components include: the RFID Tags, the Readers, the EPC Middleware, the EPC Information Services (EPCIS), the Object Naming Service (ONS) and Discovery Services. The specifications define how Readers interrogate an RFID tag. The Middleware filters and processes data gathered by Reader components. Data are then stored in EPCIS repositories and made available to external clients via the EPCIS Query Interface. The ONS offers a name resolution service that translates an EPC code into the URLs pointing to the EPCIS repositories storing data about that EPC.

The EPCIS specifications define a distributed information system enabling the exchange of EPC-related data across several enterprises [6]. EPCIS handles two types of data: *Event Data*: events that occur during the execution of business and logistic processes. They can be acquired via the EPCIS Capture Interface and made available to external clients via the EPCIS Query Interface. Event data represent state changes of the entities of interest (e.g., “EPC X has been detected in Location L at 13:23 on March 15, 2012”).

- *Master Data*: data that define the context needed to correctly interpret acquired event data (e.g. “Location L refers to the Warehouse located in Marconi Street 10, Firenze, Italy”). They can be made available through the EPCIS Query Interface and acquired through custom interfaces.

A. EPCIS Reference Scenario

Let us consider the following reference scenario: a product tagged with an EPC code is ready for the delivery in stock. When the product crosses a gate to start the delivery stage, an EPCIS ObjectEvent is generated. According to the EPCIS vocabulary, this phase is described as “shipping”. A similar event is generated when the product is loaded onto a mean of transportation (“loading”), unloaded (“unloading”) and delivered to the recipient (“receiving”). These events occur in different organization domains, and thus, are typically stored in different EPCIS repositories. Nonetheless, these Object Events are interrelated (e.g., they pertain to the same EPC or list of EPCs) and have logical relationships with other data and events (e.g., Master data that characterize the locations and business steps encoded in the above mentioned Events, and Events for other EPCs that arrived in the same location at a certain instant of time).

It is possible to conceive many applications that access and meaningfully organize this information to provide value-added services. For instance, a logistic provider could supply his customers with a web application able to track the product through the different stages of transport (e.g., by showing both the planned and the tracked route on a map); Public safety organizations could monitor the transport of dangerous goods or special waste; Public Administrations could build statistics on economic activities and transport flows on a given geographical area. We argue that developers of applications accessing EPCIS data could be supported in their tasks by mechanisms that explicitly represent logical relations among Events and Master Data and allow navigating these relations seamlessly with respect to organization domains.

To achieve these objectives our approach relies on REST and Linked Data principles, which are briefly explained in the next Section.

III. Background

In this Section we introduce the main principles of Linked Data and REST paradigms.

A. Linked Data

Linked Data is an emerging concept that promotes a paradigm shift in how information is modeled and accessed to turn the Web into a platform for data and information integration and composition. To this end, complex information may be built aggregating simpler information units that are globally addressable and linkable.

In 2006 Bizer, Berners Lee et al. published the Linked Data principles [10]. These principles are best practices for publishing data "*in a way that all published data become part of a single global data space:*

1. *Use URI as names for things.*
2. *Use HTTP URIs so that people can look up those names.*
3. *When someone looks up a URI, provide useful information, using standards (RDF, SPARQL).*
4. *Include links to other URIs, so that they can discover more things*" [10].

In the first principle the term "things" is used, but "resources" would be more appropriate. Indeed, Uniform Resource Identifiers (URIs) [11] are the recommended identifiers. In the second principle, a constraint is put on the URI scheme, since HTTP URIs can be dereferenced (i.e. used as reference to retrieve the resource representation) by exploiting existing Web technologies and the HTTP protocol [16]. The third principle states that the information the URI refers to has to be meaningful and useful, and properly represented through standard languages, suggesting for this purpose the use of RDF and SPARQL specifications. Finally, the fourth principle specifies that relations among resources have to be explicitly represented through links to support browsing and discovery.

The emerging Linked Data paradigm leads towards a global browsable information space where data from different information sources are connected and aggregated to form new information, opening up at the same time new possibilities for domain-specific applications [10].

Most Linked Data initiatives typically focus on data sharing in the context of the publicly accessible Web, thus contributing to the continuous growth of a web of globally linked open datasets. Although we argue that this emerging web of linked data is based on principles that emphasize the "System of Systems" evolutionary and open nature of the Web [17], these principles can be equally well applied to data that exist in private domains or that "*straddle the public and the private*" [18]. This is especially the case of applications in the RFID and EPC domains, where, depending on the type and relevance of data, different data sharing policies may apply (e.g. private, open, federated repositories or a combination of them).

B. REST

The REST architectural style relies on the main concept of "resource", which is an abstract information entity. On the REST vision, data sets and objects handled by client-server application logic are modeled as resources. Clients and servers exchange information declining resources into concrete representations and interact with them through a uniform interface, i.e. a predefined set of operations with defined and shared semantics. Although REST is not tightly bound to any protocol, HTTP is widely adopted for its implementation. The key principles of REST are fourfold [9]:

1. *Use URIs to identify resources*. Resources are exposed using URIs. URIs belong to a global addressing space and so resources identified with URIs have a global scope.
2. *Adopt a uniform interface*. The interaction with the resource is fully expressed with four primitives: create, read, update and delete. In HTTP-based implementations, they are mapped on the PUT, GET, POST and DELETE verbs: GET gets the resource state; PUT sets the resource state; DELETE deletes a resource; POST extends a resource by creating a child resource [19].
3. *Adopt self-descriptive messages*. Each message contains the information required for its management. For instance, metadata can be used for content negotiation (i.e. negotiate the format of the representation), errors notification, etc.
4. *Adopt stateless interactions*. Each client request must contain all of the information that are required by the server to understand the request. Session state is kept by the client and no client session data are stored in the server [9]. Instead, the server manages and stores the state of the resources it exposes.

The adoption of REST over HTTP has a very low cost since it leverages well-known W3C/IETF standards and the required infrastructure (i.e., HTTP clients and servers) is widespread. Moreover, the statelessness property makes a RESTful server scale well with the number of clients and eases the adoption of load-balancing policies. Performance optimization can be achieved by choosing lightweight message formats [19].

IV. Related Work

Linked Data principles have been applied in several projects promoted by public and academic communities [10] while Linked Data for industrial applications is a matter of ongoing research [20]. Servant in [21] argues that a company's information systems can be envisioned as a space of Linked Data. Analogously, works in [22], [23] discuss the adoption of Linked data principles and reference technologies to enable data sharing and access to internal and external repositories within and across enterprises boundaries.

Logistics and supply chain scenarios that exploit RFID and EPC technologies pose challenging requirements for data exchange and interoperability. In this domain, Guinard et al. [24] extend the existing EPCIS repository with a module offering a RESTful interface on top of standard WS* Query Interfaces. REST has also been applied in several works in the "Internet of Things" [25] and "Web of things" domains [26], [27]. Pfisterer et al. [27] propose semantic entities as a concept to map sensors and their raw output to real-world entities with high-level states. They also introduce the notion of "virtual sensors", i.e. RESTful web services that expose operations for changing the state of these real-world entities. Colitti et al. [26] exploit the REST guidelines to develop a web application allowing the user to browse measurements gathered by Wireless Sensor Networks.

A. Motivation of our work

Our approach is close to that of Guinard et al. [24], in that we aim at exploiting Web-derived paradigms for enabling EPCIS data sharing across enterprises. But while the work proposed by Guinard [24] focuses on REST principles, our contribution aims at exploiting both REST and Linked Data principles.

REST and Linked Data are two recent paradigms and while both have roots on the Web their mutual relation is a matter of study. Though REST and Linked Data show similarities (e.g., the resource abstraction, the use of URIs for resource identification, and the need of representing relations between resources), they have different scopes: REST defines an API for programming applications based on the HATEOAS (Hypermedia as the Engine of Application State) [9] constraint, while Linked Data focuses on the definition of a distributed data model. We agree with the analysis carried out by Page et al. [28], who highlights how these scopes could complement each other in the design of domain-driven applications: on one side Linked Data proposes principles for defining a shared representation of data in a given application domain, on the other side REST defines a lightweight API for accessing, modifying and publishing such data according to the given application domain purposes.

By exploiting both paradigms, EPC events can be handled as a graph of globally-addressable information resources that can be navigated, queried, and aggregated through a uniform interface and seamlessly across organization domains.

While in [20], [21], [22], [23] the adoption of Linked Data principles is achieved by relying on RDF and SPARQL technologies, a more general interpretation of Linked Data is provided by Wilde et al. [29], who define Linked Data as "the general concept of publishing interlinked data representations, without referring to the one specific way of implementing it that is often associated with that term as well". This point of view is shared also by Davis [30] who proposes a set RESTful services that return inter-linked Atom-based resource representations. Analogously, in this work we adopt the definition of Linked Data proposed by Wilde et al. in specifying an information model that is not dependent on any specific implementation technology (e.g., RDF and SPARQL), although it can be enhanced with semantic annotations.

In the following section we introduce the InterDataNet Framework, which has been conceived by exploiting Linked Data and REST principles.

V. InterDataNet

InterDataNet (IDN) [14] is an open source framework offering capabilities for representing and managing information units and their structural and semantic relations on the Web, in a RESTful way. For the sake of conciseness, in this paragraph we provide a brief introduction of the IDN framework. Further details can be found in [14].

The main goal of IDN is to enable the easy exploitation and reuse of globally web-addressable information units to support collaboration around data. For this purpose, IDN considers documents as first class entities. A document can be defined as "a set of information pertaining to a topic, structured for human comprehension, represented by a variety of symbols…" [31]. Humans are thus very comfortable to conceive data aggregation in documental form. Therefore, in IDN, documents represent a structured aggregation of data that conveys some meaningful (and shared) information in a given application domain. In the following, we refer to a document in IDN as IDN-Document.

A. Information Model

The IDN-Information Model defines the rules for organizing data in an IDN-Document.

Definition 1. An IDN-Document is a directed graph G = (V, E) where V is the set of vertices and E is the set of edges. The elements of V and E are the nodes containing the granular information (IDN-Nodes) and the relations between IDN-Nodes, respectively. IDN supports two types of relations between IDN-Nodes: aggregation (i.e., containment) and reference.

Definition 2. An *IDN-Node* is a set $S = \{C, P\}$, where C is the set of content elements (i.e., data) and P is the set of properties (i.e., metadata) that characterize C.

Definition 3 The Aggregation Link represents a container-content relationship. The node where the edge starts from aggregates and therefore contains the node the edge points to.

Definition 4 The Reference Link represents a pointer towards the referred resource. No further meanings are associated with the Reference Link. To better understand the Reference Link role, it could be somehow compared with the HTML "href" attribute.

Let D be an *IDN-Document* modeled as a graph G. Hence the topology of G expresses the *IDN-Information Model* used to represent D. Through the *IDN-Information Model* it is possible to define an IDN-Document as an aggregation of data provided by different information sources. Indeed, an IDN-Node can be referred to by more than one IDN-Document, thus favoring the reuse of information across different applications (Fig. 1). This is possible since each IDN-Node is associated to an information provider that is authoritative for

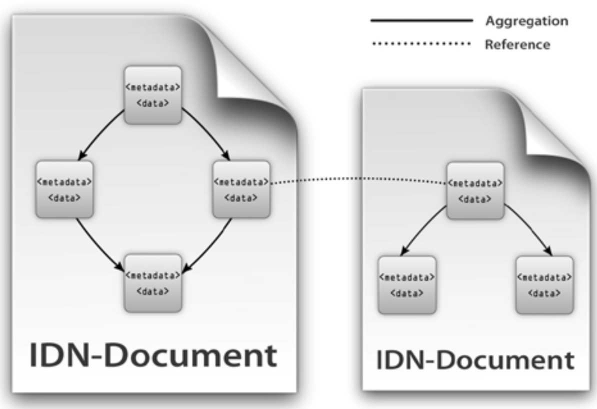

Fig. 1. IDN Information Model

the information the IDN-Node refers to. Hence, gathering information from the proper sources enforces an appropriate responsibility distribution across the information providers, who are responsible for the quality of the provided information.

In addition, IDN-IM can be extended with metadata enforcing privacy, licensing, security, provenance, consistency, versioning and availability properties attached to IDN-Nodes and affecting IDN-Documents [14]. Such features are crucial to support effective and trusted collaboration on real world scenarios.

B. Service Architecture

IDN-Documents are exposed as resources through the IDN-Service Architecture (IDN-SA) API. The IDN-SA API is a set of generic REST interfaces for addressing, resolving and handling IDN-Documents and IDN-Nodes.

IDN-SA (Fig. 2) is the architecture that implements the services needed to enforce the IDN-Document's properties and capabilities. IDN-SA has been designed with separation of concern and information hiding principles in mind and it is organized according to a layered architectural pattern. IDN-SA has four main layers, from the top to the bottom: Virtual Resource (VR), which provides RESTful APIs for accessing creating, and modifying IDN-Documents; Information History (IH), which implements information versioning capabilities; Replica Management (RM), which guarantees resources availability through resource replication across distributed hosts; Storage Interface (SI), which offers persistence capabilities. In addition, we defined a set of horizontal services, including names resolution and security and privacy management.

By exploiting IDN-SA APIs (namely the Virtual Resource upper interface), it is possible to develop web-based applications (called IDN-Compliant Applications, IDN-CA) that browse and handle graphs of distributed information units (i.e. IDN-Documents), through a uniform interface.

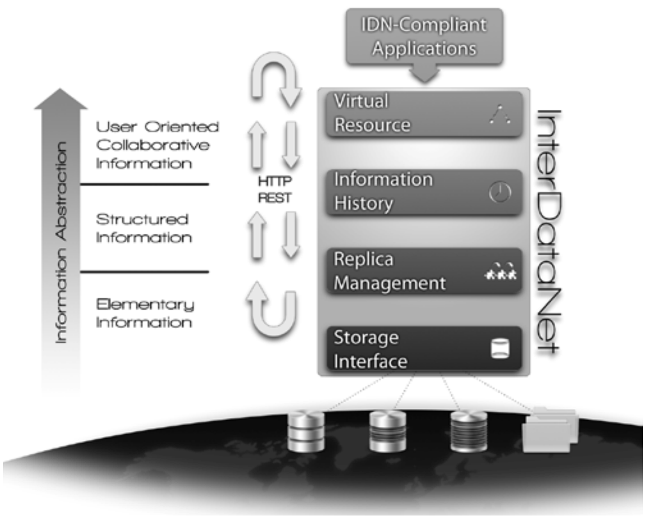

Fig. 2. IDN Service Architecture (IDN-SA).

C. Search Service

The IDN SEarch Service (ISE) is the component that offers search capabilities to retrieve IDN-Documents. ISE is made of two components: a Controller and a Search Engine. The Controller is an HTTP manager that exposes a REST-based search interface. It manages the HTTP requests coming from the Virtual Resource (VR) services and client applications (IDN-CA). The Search Engine implements a data structure (i.e., a distributed inverted index) for handling client queries and performing information retrieval. For its implementation, we chose Apache Solr [32] because it is an open source solution that shows good performances and is well-documented.

In this context, Apache Solr communicates only with the Controller through HTTP messages. In other words, the Controller manages the interactions with the IDN-CAs and VRs (the queries are submitted through HTTP requests). Then, the Controller forwards specific HTTP requests to Apache Solr for the information retrieval and finally the Controller processes the response returned by Solr and delivers the results to the client.

Fig. 3. shows a Search Service deployment example. It is possible to change the number of Search Service components and the topology of the network by configuring specific ISEs parameters like the connected neighbor ISEs and the registered VRs.

Each ISE offers a search service over the information that is handled by registered VRs. When one or more information nodes handled by a VR are created or modified, the VR notifies the corresponding ISEs using HTTP PUT messages. In order to perform HTTP requests that alter the state of the indexed resources a VR must be registered to the authoritative ISE. In the example shown in Fig. 3 VR1 and VR2 are registered to ISE1, VR3 is registered to ISE2, and VR4 and VR5 are registered to ISE3.

The HTTP message bodies sent to ISEs are XML documents containing information about the resources to be indexed arranged in a list whose elements are described as follows:

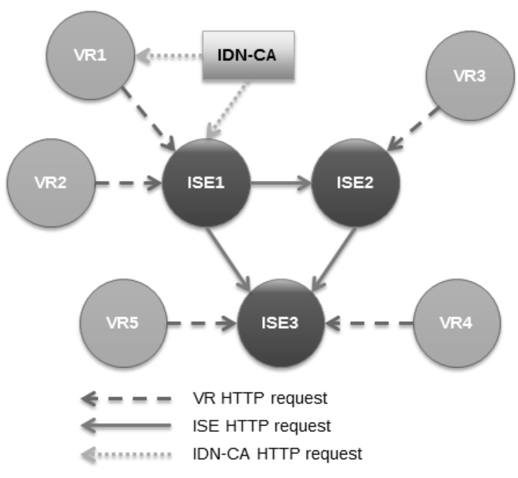

Fig. 3. IDN Search Service.

- <id>: identifier of the resource to be indexed. The VR level resource URI is used;
- <time>: the timestamp optionally associated to the resource (in millisecond from 1/1/1970);
- <keywords>: arbitrary list of keywords (<keyword>) optionally associated to the resource. The keywords are the critical information for the retrieval procedure. The more keywords are provided for a resource, the easier it will be for the resource to be retrieved while the pertinence of a keyword to the subject affects the quality of the retrieved data.

The information retrieval phase triggers the propagation of different HTTP GET requests across the ISEs network. Requests contain the client query and they are exchanged through the deployed Search Service providers. As shown in Fig. 3, the query request coming from the IDN-CA to the ISE1 is propagated to ISE2 and ISE3; each of these recursively performs the query propagation. Let's suppose that the HTTP message propagation time is uniform in the ISE network. Then ISE3 at a different time will receive the same request twice. Since a request is marked with an identifier, ISE3 will reject the request coming from ISE2 because it already handled it. Finally, the propagated query returns to the original client as an overall response each ISE contributed to.

A search request is submitted by invoking an HTTP GET operation on a URI with the following syntax:

```
http://net_location:port/ISE/index/${keywords}[§{time}][@{domain}]
```

- ${keywords}: the $ special character is used to identify a search expression. The expression is a concatenation of keywords put in relation through logic operators. For example a valid expression is (feature_1||feature_2)&&(!feature_3) that stands for (keyword_1 OR keyword_2) AND (NOT keyword_3);
- §{time}: the time expression is optional and it is used to filter a query on the basis of temporal attributes. For example, a valid time expression is (time>135331596295)&&(time<=135331597000) that stands for (time>135331596295) AND (time<=135331597000);
- @{domain}: the optional domain expression implements an additional filter by specifying the resource domain (or a part of the domain). For example a valid domain is domain_1/domain_2: only resources that contains the sub-path domain_1/domain_2 into their <id> tag are retrieved.

Each HTTP GET response returned to the client contains an XML document representing the results list. Each item in the list contains qualifying information about the corresponding indexed resource. In addition, all items have a <score> tag representing the pertinence to the processed query (the <score> value is evaluated by Apache Solr internal algorithms).

VI. Information-centric Approach for EPCIS

The EPC scenario is a challenging data consumption and production environment. For each tagged item, vast amount of information is entered into the system in the form of events. Events describe the relevant actions related to an EPC at a certain time. The EPCIS specification [6] defines four types of events: 1) an Object Event is a general event related to an object. 2) An Aggregation Event describes the aggregation of more EPCs together inside a parent EPC container. For example, the actions of gathering a number of RFID tagged cases and loading them onto a pallet produce an Aggregation Event. 3) A Quantity Event describes an event pertaining to a number of elements of an object class where the individual instances are not identified. This type of event can be used to report inventory levels of a product. Finally, 4) a Transaction Event describes the association or disassociation of an object to a specific transaction.

Events establish relations between objects, contexts and sets of master data. For example an event can describe the loading phase of a pallet of microwave ovens on the currier truck occurred in a certain location. If the location is the factory facility, it will be likely to find it replicated in many other events. Due to their nature, master data are eligible to be frequently reused. This scenario could considerably benefit a graph data structure reflecting the abstract information relations. Therefore, the location of the previous example could be represented as a graph node the other entities (for example, events) connect with.

These considerations led us to propose the IDN approach to manage EPCIS information. Our intent is to build a browsable graph of individually addressable EPCIS data to support information reuse and efficient exploitation [15].

To this purpose, a "proof of concept" web application has been designed and implemented. The application is called "EPCWeb" [33] and is focused on two key assets: 1) to enable a web centric approach offering RESTful APIs and 2) to enable a data centric approach building and exploiting a global graph of related resources.

Therefore, the EPCWeb web application is an EPCIS Capture Interface and Query Interface implementation which converges on the aforementioned assets.

A. IDN-IM

The EPCWeb application supports all the event types defined in the EPCIS specification. In compliancy with it, the four event types derive from a general event entity which carries the common characteristics. A domain analysis highlights that all the events may have an event time, a record time, a time zone offset, a disposition, a read point, a business location and a business transaction list. Three events out of four are strongly related to one or more EPC elements. These information entities have been divided in two groups depending on how likely they were to be reused. The ones more likely to be reused were candidate to be IDN-Documents while the others (with more descriptive nature) were candidate to be attributes.

Fig. 4-7 show the IDN-IM for each EPCIS event type.

In Fig. 4-7 the bold arrows represent the Aggregation Links while the dotted arrows represent the Reference Links. Not to weight down the pictures readability, the attributes are omitted but they are listed in Table I (the R and O placeholders stands for “required” and “optional”, respectively).

TABLE I
ATTRIBUTES PER EVENT TYPE (R = REQUIRED, O = OPTIONAL)

Event / Attr.	Object	Aggregation	Transaction	Quantity
eventTime	R	R	R	R
zoneOffset	R	R	R	R
epcClass				R
Quantity				R
Action	R	R	R	
bizStep	O	O	O	O
disposition	O	O	O	O
transactionList	O	O	R	O

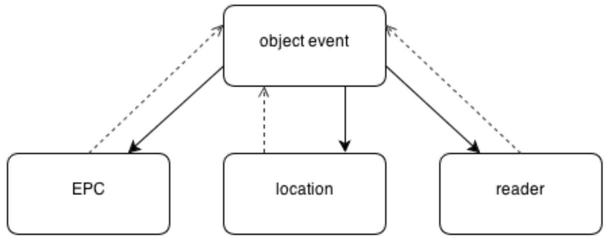

Fig. 4. Object Event IDN-IM.

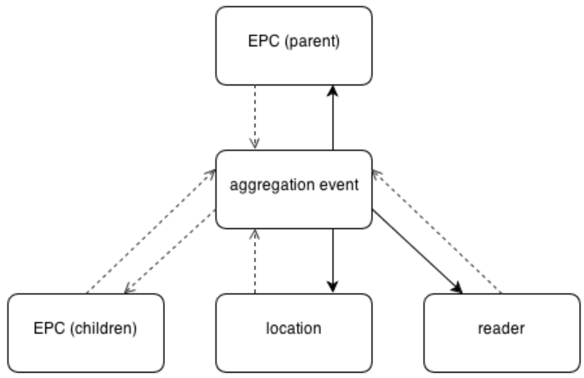

Fig. 5. Aggregation Event IDN-IM.

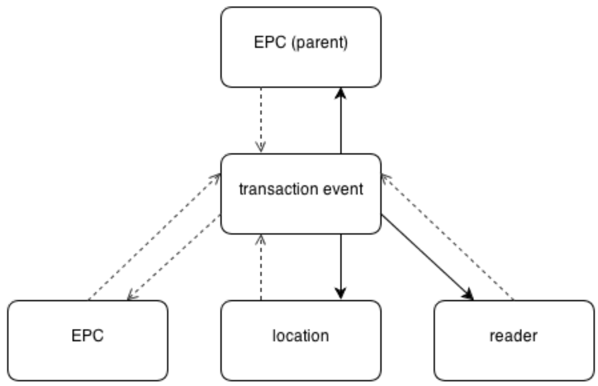

Fig. 6. Transaction Event IDN-IM.

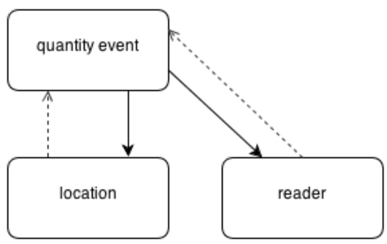

Fig. 7. Quantity Event IDN-IM.

B. *Capture Interface*

The standard EPCIS Capture Interface acquires EPCIS events to populate an Event Repository. The proposed RESTful Capture Interface accepts a POST request whose payload is the current standard XML representation of an EPCIS event. This request triggers the creation of the corresponding IDN-Document in the IDN infrastructure. The POST response returns an HTTP URI referencing the newly created resource. In compliancy with the aforementioned data reuse key asset, when an event is POSTed to the Capture Interface, the system is queried for the related entity elements (EPC, location or reader) to be processed. If an element is found it is reused, otherwise it is created and indexed by the ISE service. For data having a URN associated, the URN is used as a keyword in the indexing procedure so that it is always possible to retrieve the HTTP URI from the URN. In this sense, the ISE serves as the ONS [6]. The URI templates for the resources are listed hereafter:

- URI template for Event resources:
  `http://authority/event/{event_id}`
- URI template for Reader resources:
  `http://authority/reader/{reader_id}`
- URI template for Location resources:
  `http://authority/location/{location_id}`
- URI template for EPC resources:
  `http://authority/epc/{epc_id}`

where `authority` refers to the authoritative VR service for the IDN-Document addressed in the URI. Each player in the reference scenario can choose its own trusted VR service to be authoritative over the data produced within its domain.

C. *Query interface*

The standard EPCIS Query Interface allows querying an EPCIS Repository. Thus, the proposed RESTful Query Interface allows to easily query the IDN-based EPCIS repository. The query is submitted via a GET request on an HTTP-URI compliant with specific URI templates. The URI templates available for querying reflect the natural structure of the IDN-Documents enabling a simple and logic retrieving paradigm. Examples of queries are listed in the following:

- which events are related to a given location (location_id) at time t?
`http://authority/location/location_id/event?time=t`
- which events are related to a given reader point (reader_id) at time t?
`http://authority/reader/reader_id/event?time=t`
- which readers have detected a given EPC (epc_id) at time t?
`http://authority/epc/epc_id/event/reader?time=t`

where authority refers to the IDN-CA. The result is an XML response message whose payload contains a list of EPCIS elements instances that satisfy the query.

D. Test Results

The EPCWeb application has been tested in a LAN environment to check the performance under different conditions. More precisely, the test architecture is composed by an instance of the EPCWeb application deployed on a machine, an IDN stack deployed on a different machine and two Search Services, deployed on a single machine each.

Tests were run on PCs equipped with an Intel Core 2 Duo processor (2.4 GHz) and 4GB DDR2. The application server adopted is Apache Tomcat 6 [34] with the JDK 1.7 [35] development environment.

The testing scenario inspected two significant cases: the submission of a list of events to the Capture Interface and the retrieval of a set of results through the Query Interface. In the trials have been used payloads of different size to test the application under different usage conditions.

The events submission scenario analyses the worst case where every information is missing in the system and it has to be created instead of reused. The payload condition testing has been performed submitting lists of 3 and 10 events to highlight the response of the application while handling different loads. As expected the payload size affects significantly the overall application performances.

The same has been done with the retrieval scenario: it has been executed a query to resolve 22 elements and a query to resolve a single element. As expected, the number of results does not affect significantly the performance of the system.

The test cases have been performed running fifty trials each. The average results are listed in Tab. II and Tab. III and refer to the events submission scenario and the results retrieval scenario respectively. The No. of created IDN-Nodes column in Table II reports the number of IDN-Nodes generated as a side effect of the events submission accordingly to the IDN-Document creation strategy discussed in the paragraph VI. A.

TABLE II
SUBMISSION TEST RESULTS

Action	No. of events	No. of IDN-Nodes created	Time (ms)
POST event	3	12	4742
POST event	10	40	15798

TABLE III
RETRIEVING TEST RESULTS

Action	Retrieved results	Time (ms)
GET location	1	266
GET event	22	390

VII. CASE STUDY

EPCWeb is a Java Web Application developed with the Spring 3 framework [36] for the server side and HTML5 and JQuery 1.8 [37] Javascript library for the client side. It has been conceived as a proof of concept to test the effectiveness of the proposed approach. To this purpose, it is shipped with an online data explorer to support a visual discovery and browsing of EPC related data.

Through the data explorer user interface it is possible to test the Query Interface composing the URL as explained in paragraph VI.C. The results are presented as a list of clickable URIs. Clicking a URI it is possible to trigger an AJAX request to retrieve the IDN-Document the URI refers to. When the IDN-Document is returned, it is arranged in the layout and presented to the user. When a document is displayed, data are divided in information about the EPCIS entity and links towards other EPCIS entities which are the Aggregation Links and the Reference Links (towards other IDN-Documents) discussed in the section V.A.

The information about other EPCIS entities is presented as formatted text, while links are kept clickable to support a complete browsing experience across the graph of EPC data.

The EPCWeb application is available at the url: http://idn.det.unifi.it:20282/EPCWeb/

VIII. CONCLUSION

In this paper we proposed an implementation of the EPCIS specifications based on the adoption of Linked Data and REST principles.

As a matter of fact, expected benefits of such principles include the development of scalable applications and easy sharing of data and models across the Web. We discussed the adoption of such principles in the EPCIS domain, with the objective of improving information sharing across organizational boundaries and easing the development of web-based applications exploiting data and events made available through the EPC Network. These principles have been put into practice through the use of the InterDataNet (IDN) Framework.

Coherently with the IDN design methodology, we discussed the design of an Information Model and RESTful services compliant with the EPCIS data model and interface specifications.

Finally, future work will include an improvement of the IDN performance and an enhancement of the EPCWeb application with improved search and discovery capabilities through the adoption of semantic annotations [38] [39] [40].

ACKNOWLEDGMENT

The authors thank Mr. Luca Capannesi for his technical support.

REFERENCES

[1] D. Guinard, V. Trifa, F. Mattern, and E. Wilde: "From the Internet of Things to the Web of Things: Resource-oriented Architecture and Best Practices", Architecting the Internet of Things, 2011, pp. 97-129.
[2] F. Paganelli and D. Parlanti: "A DHT-Based Discovery Service for the Internet of Things", Journal of Computer Networks and Communications, vol. 2012, 2012. doi:10.1155/2012/107041.
[3] S. Evdokimov, Sergei; F. Benjamin, S. Kunz, N. Schoenemann: "Comparison of Discovery Service Architectures for the Internet of Things", in Proc. of the 2010 IEEE International Conf. on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), pp. 237-244, 7-9 June 2010.
[4] I. Zappia, F. Paganelli, D. Parlanti: "A Lightweight and Extensible Complex Event Processing System for Sense and Respond Applications", Expert Systems with Applications 39 (12), pp. 10408-10419, 2012. Available: http://dx.doi.org/10.1016/j.eswa.2012.01.197
[5] EPCglobal: F. Armenio, H. Barthel, et al. The EPCglobal Architecture Framework, 2009.
[6] EPCglobal, EPC Information Services Ver. 1.0.1 Specification.
[7] A. P. Barros, and M. Dumas: "The Rise of Web Service Ecosystems", IT Professional, vol.8, no.5, 2006, pp.31-37.
[8] J. Hendler, N. Shadbolt, W. Hall, T. Berners-Lee, and D. Weitzner: "Web science: an interdisciplinary approach to understanding the web", Comm. ACM 51, 7, 2008, pp. 60-69.
[9] R. T. Fielding: "REST: Architectural Styles and the Design of Network-based Software Architectures", Doctoral dissertation, University of California, Irvine, 2000.
[10] C. Bizer, T. Heath, and T. Berners-Lee: "Linked data - the story so far", Int. Journal on Semantic Web and Information Systems, 5(3), 2009, pp. 1-22.
[11] T. Berners-Lee et al. (2005). Uniform Resource Identifier (URI): Generic Syntax. RFC 3986. Available: http://tools.ietf.org/html/rfc3986.
[12] Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recom., 2004. Available: http://www.w3.org/TR/rdf-concepts/
[13] SPARQL Query Language for RDF, W3C Recomm., 2008. Available: http://www.w3.org/TR/rdf-sparql-query/
[14] M.C. Pettenati, L. Ciofi, F. Pirri, D. Giuli: "Towards a RESTful Architecture for Managing a Global Distributed Interlinked Data/Content/Information-Space", Future Internet: Achievements and Promising Technology, Springer, 2011.
[15] S. Turchi, L. Ciofi, F. Paganelli, F. Pirri, D. Giuli: "Designing EPCIS through Linked Data and REST principles", in the Proc. of 2012 International Conf. on Software, Telecom. and Computer Networks, SoftCOM 2012, Split, Croatia, Sept. 2012.
[16] R. Fielding, et al. (1999). Hypertext Transfer Protocol - HTTP/1.1. RFC:2616. Retrieved June 14, 2009. Available: http://www.w3.org/Protocols/rfc2616/rfc2616.html
[17] M. W. Maier, "Architecting principles for systems-of-systems", Systems Engineering, vol. 1, 1998, pp. 267–284.
[18] T. Heath , C. Bizer: "Linked Data: Evolving the Web into a Global Data Space", 1st ed., Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan & Claypool, 2011.
[19] C. Pautasso, O. Zimmermann, and F. Leymann: "Restful web services vs. big web services: making the right architectural decision". In Proc. of the Int. Conf. on World Wide Web, 2008.
[20] B. Hyland: "Preparing for a Linked Data Enterprise", Linking Enterprise Data, Springer, 2010, 51-64.
[21] F. P. Servant: "Linking Enterprise Data", In Linked Data on the Web Workshop at the 17th Int. World Wide Web Conf., 2008.
[22] B. Hu, G. Svensson: "A Case Study of Linked Enterprise Data", In Proc. of the 9th int. semantic web conf. on The semantic web – Vol. Part II, Springer-Verlag, Berlin, Heidelberg, 129-144.
[23] M. Graube, J. Pfeffer, J. Ziegler, L. Urbas: "Linked Data as integrating technology for industrial data", in Proc. of 14th Int. Conf. on Network-Based Information Systems, 2011, 162-167.
[24] D. Guinard, M. Müller, J. Pasquier: "Giving RFID a REST: Building a Web-Enabled EPCIS", Proc. of the Internet of Things Int. Conf. (IoT 2010), 2010, Tokyo, Japan.
[25] V. Stirbu: "Towards a RESTful Plug and Play Experience in the Web of Things". In Proc. of the 2008 IEEE Int. Conf. on Semantic Computing (ICSC '08), Washington, USA, 512-517.
[26] W. Colitti, K. Steenhaut, N. De Caro, B. Buta, e V. Dobrota: "REST Enabled Wireless Sensor Networks for Seamless Integration with Web Applications", in Proc. of the 2011 IEEE 8th Int. Conf. on Mobile Adhoc and Sensor Systems (MASS), pp. 867–872, 17-22 Oct. 2011.
[27] D. Pfisterer, K. Romer, D. Bimschas, O. Kleine, R. Mietz, C. Truong, H. Hasemann, A. Kro ller, M. Pagel, M. Hauswirth, M. Karnstedt, M. Leggieri, A. Passant, e R. Richardson: "SPITFIRE: toward a semantic web of things", IEEE Communications Mag., vol. 49, no. 11, pp. 40–48, Nov. 2011.
[28] K. R. Page, D. C. De Roure, and K. Martinez: "REST and Linked Data: a match made for domain driven development?", In Proc. 2nd Int. Workshop on RESTful Design. ACM, 22-25.
[29] E. Wilde, E. Kansa and R. Yee: "Web Services for Recovery.gov", School of Information, UC Berkely, CA, Report 2009-035, Oct. 2009. Available: http://escholarship.org/uc/item/0fv601z8.
[30] C. Davis: "Achieving Linked Enterprise Data with RESTful Services and Link Relations" presented at the 2011 W3C Workshop on Linked Enterprise Data Patterns Data-driven Applications, Cambridge, MA, USA.
[31] R. H. Sprague Jr.: "Electronic Document Management: Challenges and Opportunities for Information Systems Managers", MIS Quarterly, 19(1), 1995.
[32] Apache Solr, Official Web Site. Available: http://lucene.apache.org/solr/index.html
[33] EPCWeb web application. Available: http://idn.det.unifi.it:20282/EPCWeb/
[34] Apache Tomcat, Official Web Site. Available: http://tomcat.apache.org/
[35] JDK 1.7, Official Web Site. Available: http://www.oracle.com/technetwork/java
[36] R. Johnson et al.: Spring Framework Reference Documentation 3.0 Available: http://static.springsource.org/
[37] JQuery. Available: http://jquery.com/
[38] F. Paganelli, D. Parlanti: "A Dynamic Composition and Stubless Invocation Approach for Information-Providing Services", IEEE

Transactions on Network and Service Management, accepted Nov. 15th 2012, (in press).

[39] F. Paganelli, D. Parlanti, D. Giuli, “Message-based service brokering and dynamic composition in the SAI middleware”, in the Proc. of IEEE 7th Int. Conf. on Services Computing (SCC 2010), Miami, FL, July 2010, pp. 474-481.

[40] D. Parlanti, F. Paganelli, D. Giuli: “A service-oriented approach for network-centric data integration and its application to maritime surveillance”, IEEE Systems Journal 5 (2) , art. no. 5643938 , pp. 164-175, 2011.

Federica Paganelli (M’07) received a Ph.D. degree in Telematics and Information Society from the University of Florence, Italy, in 2004. She is a Senior Researcher at the National Interuniversity Consortium for Telecommunications (CNIT), Italy. Her research interests include context-aware systems, Ambient Intelligence, service-oriented computing, and next generation networks

Stefano Turchi graduated in Computer Engineering (M.S.) in 2010 at the University of Florence. From January 1st 2011 he is a Ph.D. Student in “Informatics, Systems and Telecommunications” Doctoral Course. Currently he is researching in the field of Distributed Architectures, Web Applications, Data Security and Data Privacy in distributed and granular data scenarios.

Lorenzo Bianchi received the “Laurea” (M.S.) degree (cum laude) in Computer Engineering, from the University of Florence, Italy, in 2011. Since January 2012 he is Ph.D. student in “Informatics, Systems and Telecommunications”. He is presently with the Telematic Systems and Remote Sensing group at the Department of Electronic Engineering, University of Florence.

Lucia Ciofi is a Telecommunications Engineer who achieved a Post-Doc position in the Integrated Laboratory of Radar & Telematics Technology. In 2011 obtained a Ph.D. in Telematics and Information Society. Since 2008 she has taught seminars and lectures in the course of Telematics Systems for Master of Science in Computer Engineering and Telecommunication Engineering. Her main research interest are: Web Science, REST, Cloud Computing.

Maria Chiara Pettenati is Program Manager at ICON (International Center of Computational Neurophotonics) Foundation. She received the M.S. degree in electronics engineering (telecommunication curriculum in 1996). In 2000 she received the Ph.D. title in Telematics and Information Society. Until December 2011 she was researcher by the Telematics Laboratory of the Electronics and Telecommunications Department of the University of Florence. Until 2004 she held a post-doctoral research position in the same laboratory. From 1997 to 1999, she has been working at the Computer Science Department (DI - LITH Laboratory) of the (Swiss Federal Institute of Technology) EPFL.

Franco Pirri received the “Laurea” (Dr. degree) from the University of Pisa in 1971. He joined the Department of Electronics and Telecommunications of the University of Florence in 1973 and became Associate Professor of “Industrial Electronics” in 1982. He is author of more than 75 papers, congress presentations and technical reports. His professional and academic experience has been focused on digital system design, with emphasis on communication networks, multimedia applications and application specific integrated circuits (ASIC). He is member of the IEEE, the IEEE Computer Society and the IEEE Communication Society.

Dino Giuli is Full Professor in Telecommunications at the Department of Electronics and Telecommunications of the University of Florence where he was Director from 1999 until 2004. Since 1996 he is Promoter and Scientific Coordinator of the Ph.D. program in “Telematics and Information Society”. His research activities have been principally oriented in the domain of Telematics and Environmental Monitoring Systems. Dino Giuli is author of more than one hundred and fifty scientific publications, mainly published on scientific journals and international conference proceedings. He is AEI member and IEEE senior member.