Privacy-enhanced Filtering and Collection Middleware in EPCglobal Networks

Wiem Tounsi, Nora Cuppens-Boulahia, Frédéric Cuppens, Joaquin Garcia-Alfaro*
Institut Mines-Telecom, TELECOM Bretagne, France
*Institut Mines-Telecom, TELECOM SudParis, France
{*FirstName.SurName*}*@telecom-bretagne.eu* ; *joaquin.garcia_alfaro@telecom-sudparis.eu*

***Abstract*—Collection and distribution of Radio Frequency IDentification (RFID) data are subject to various privacy concerns. These concerns are of paramount importance when sensitive data are processed (e.g., medical data). Therefore, it is crucial to treat sensitive data privacy in early stages to master the data view for upper layers and to minimize, as soon as possible, the risk of unauthorized disclosures. While most recent works focus on securing the access and visibility of collected information in the final databases, data processed in the middleware do not seem involved in the process of privacy protection. Current EPCglobal standards for RFID also suffer from insufficient attention to this issue. In this paper, we propose a privacy controller module that enhances the Filtering and Collection (F&C) middleware of the EPCglobal network. We provide a privacy policy-driven model, using some enhanced contextual concepts of the extended Role Based Access Control model. The feasibility of our privacy-enhanced model is shown by integrating our solution into the F&C middleware of the Fosstrak framework, an open-source implementation of the EPCglobal network specifications.**

***Keywords*-Middleware, RFID, EPCglobal, Fosstrak, Privacy, Security**

I. Introduction

The Electronic Product Code (EPC) is an emerging technology that uses Radio Frequency IDentification (RFID) for the automatic discovery of consumer products. With the proliferation of various kinds of RFID sensors in our everyday life, there is no longer a one to one relationship between reader and application instance. The flow of data is now coming from distributed nodes towards a number of processing applications. This introduces the need of an RFID middleware to help making sense of RFID tags data by applying filters, aggregating, grouping and in some cases, adding logic to tags data captured by readers and sensors. Then, this well-formatted data is sent in real time to the backend applications for company information systems. The management of this large volume of data also adds a security need to deliver the right data to the right application and role.

While most recent works are concentrated on securing the access and visibility of the collected information in the final backend databases, which process historical events [1], [2], [3], [4], the data processed in the RFID middleware are not treated with required security properties like privacy control. A possible reason for this lack of security is that most enterprises run the RFID middleware in an internal network.

A frequent definition of privacy is *the right of individuals to determine for themselves when, how and to what extent information about them is communicated to others* [5]. Not sharing or hiding information is a fundamental definition of privacy, however, the privacy of people focuses today on who has the access to what information, rather than only what information is collected. This way, the issues of privacy are completely treated in different situations where the owner of the data chooses to share information with others. Traditional solutions for granting data privacy have been aligned on anonymizing user information or preventing personal data from disclosure by encryption solutions. However, dealing with encrypted data makes query processing expensive [6], either from a database or a middleware point of view. For example, the list of illnesses or the list of patients in a hospital, could be made publicly available, whereas the association of specific illnesses to individual patients presents a sensitive information. Thus, there is no need to encrypt both illnesses and patients lists if there are alternative ways to protect the association between them [7], [8] or to filter them in early levels.

Holding privacy issues from the lowest level of RFID data collection is important. In [9], authors proposed to handle the dimension of declared purpose in the reader to tag level. In our paper, privacy is handled in the RFID middleware level.

Numerous reasons motivate the support of privacy issues in the RFID middleware. The first reason is to master the data view before interpreting and storing it in upper layers in order to minimize, as soon as possible, the risk of unauthorized disclosures. In sensitive domains such as healthcare, embedded RFID tags attached to the patient personal devices, tools and medicines can be triggered (after the reader and tag authentication) to reply with their ID and other related information. Then, the data collected from RFID tags can potentially be acquired by multiple parties (e.g, curious applications), presumably allowing for a fine-grained and excessive surveillance mechanisms that would increasingly pervade the patients lives [10]. According to the Health Information Portability and Accountability Act. (HIPAA) guidelines [11], dealing with privacy concerns means that we must not be able to identify the product type by simply reading the RFID tag's ID code. Thus, the identification could be realized only by considering some parts of the ID code such as a product manufacturer or a family of tags depending on the requester purpose. If this control task could be held before the generated events are sent to upper layer applications, it would be interesting that an RFID middleware consider these consumer concerns before information is interpreted and stored. To illustrate this issue, let us consider two distinct applications of two services

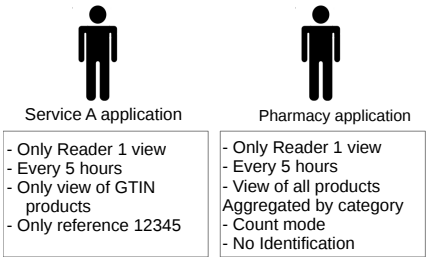

Fig. 1. Different views for applications in the same hospital

in the same hospital (a service A and a pharmacy service) wanting to receive data about a patient (cf., Figure 1). The pharmacy application is only allowed to view the total tag count depending on the requester purpose, while the service A application is allowed to view the identity of tags but only of product "GTIN" of reference "12345". In this case, we propose to handle the privacy policy before the generated events are sent to upper layer applications and databases.

The second reason for treating privacy issues in the middleware stage is that filtering data at this stage relaxes the event collection, leading to an appropriate adaptation of the queries executed by the reader over the air interface. This also allows readers to use efficiently the limited bandwidth, e.g., targeting a particular tag population or switching off some readers to make the communication bandwidth available to other readers.

Finally, as RFID communication protocols support dedicated privacy enhancing features [12], the RFID middleware will also need to support their use. For instance, to use the *kill*-command specified in EPC Generation 2 standard, the RFID middleware must provide the appropriate *kill*-passwords to the appropriate reader to apply it to the tag.

Our contributions in this paper can be summarized as follows. First, we propose to add a privacy controller to the Filtering and Collection middleware that extends existing specifications of the EPCglobal network. The feasibility of our solution is proved by integrating it to the open-source software for track and trace Fosstrak [13]. Fosstrak is a recommended implementation of the EPCglobal interfaces. Second, we provide a policy-driven solution for the EPCglobal middleware while ensuring the following features: (i) enforcing a privacy policy without interfering with the standard middleware interface (ii) using an existing privacy-aware model to store and manage privacy policy preferences, and (iii) taking into account the purpose dimension as a privacy requirement.

Paper organization. Section II describes the EPCglobal network and the main modules of the Fosstrak platform. Section III details the middleware interfaces specified by EPCglobal. Section IV introduces our privacy enhancement solution in the Filtering and Collection middleware applied on Fosstrak. Section V discusses related work. Section VI concludes the paper.

II. EPCglobal Network

The EPCglobal network, initially proposed by the MIT's Auto-ID Center [14] and further developed by members of the joint-venture EPCglobal, is currently one of the predominant standardization efforts of the RFID community. The EPCglobal network is a set of global technical standards aiming at enabling automatic and instant identification of

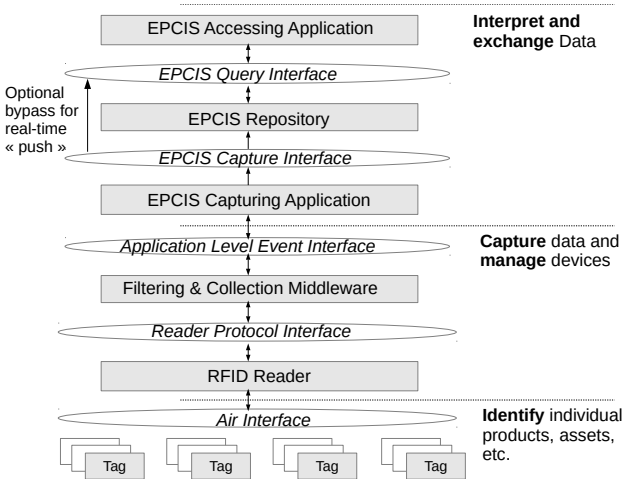

Fig. 2. EPCglobal Network components: Roles and Interfaces

individual items and sharing this information throughout the supply chain. In the following, we first present the principal components of the EPCglobal network architecture, then describe the Fosstrak platform.

A. EPCglobal Network Components

Figure 2 shows the relationship between several EPCglobal standards from a data flow perspective. Each processing step, played by hardware or software components of a typical system architecture, is mediated by an EPCglobal standard interface, from the bottom of the figure to the top:

- RFID Tag consists of a microchip that contains a unique number designating a physical item. In the case of EPCglobal standard, this number is the Electronic Product Code. EPC comprises four parts: the header, the product manufacturer, the product type, and the serial number.
- RFID Readers make multiple observations of RFID tags, when these tags are in the read zone. The observations are performed via the air interface protocol typically the UHF (Ultra High Frequency) Class1 Gen2 in the case of passive EPC tags, and are transferred to the middleware.
- Reader Interface defines the control and delivery of tag read from Readers to the middleware role. Events at this interface say: "Reader A saw EPC X at time T".
- Filtering & Collection Middleware (F&C) decouples readers and applications and provides additional aggregation and filtering functionalities. It filters and collects raw tag data over time intervals in a report, delimited by events and defined by the EPCIS (EPC Information Services) Capturing Application. It brings the concept of logical readers (i.e., addition of different readers) making data intelligible for the information system.
- Application Level Event Interface defines the control and delivery of filtered and collected tag data from F&C role to the EPCIS Capturing Application role. This allows applications to define standing query and later, to subscribe to it. Events at this interface say: "At Location L, between time T1 and T2, the following EPCs were observed".
- EPCIS Capturing Application is a F&C middleware client that supervises the operation of the lower elements, and provides business context. e.g., It checks for exceptional

conditions and present information to a human operator.

- EPCIS Capture Interface delivers the data to enterprise-level roles, including Repositories and Accessing Applications. Events at this interface say: "At location X, at time T, the following contained objects were verified as being aggregated to the following containing object".
- EPCIS Repository records EPCIS-level events that are generated by one (or more) Capturing Application(s), and makes them available for EPCIS Accessing Applications.
- EPCIS Accessing Application is responsible for achieving all the enterprise business processes, such as warehouse management, historical throughput analysis, aided by EPC-related data.

To summarize, three levels of data are generally processed in the RFID system: (i) a database level that processes historical events, (ii) a middleware level that processes real time events, and (iii) a low level that processes real time events to perform very detailed control over how readers and tags interact.

B. Fosstrak Platform

The EPCglobal organization defines a group of standard interfaces but does not specify a particular implementation strategy for different roles in the EPCglobal network. Fosstrak (ex Accada) [13] is an open source RFID software platform providing an EPCglobal-certified EPCIS Repository as well as Query and Capture clients. It features a number of extensions that address some of the challenges of RFID middleware design [15]. The Fosstrak platform consists of three separate modules: the reader module, the filtering and collection middleware module, and the EPCIS module. They are implemented with the corresponding interface specifications described in Figure 2. In this paper, we focus on the middleware module, the key location where begins the configuration of events to be collected and reported.

Regarding privacy issues in the middleware, the consideration of Fosstrak authors [15] is not sufficient to preserve the privacy of persons who are associated with tags, in particular, when the identity of both persons and tags are revealed. Before delving into the details of our proposal, let us present in the next section the ALE specifications and its security limitations.

III. Application Level Events Specifications

The Filtering & Collection middleware of the EPCglobal network uses a single interface to a large number of distributed readers and a large number of capturing applications that may be interested in the collected data. This interface is called the ALE (cf., Section II-A). ALE [16] provides a means for clients to specify, in a high-level, declarative way, what tag data they are interested in. It also provides a standardized format to report collected and filtered tag data independently from the tag data origin or the manner it was processed with. Finally, it abstracts the sources of tag data (e.g., Readers, Barcode scanners) into higher-level notions of "location". The Event Cycle Specification (ECSpec) is the XML type used to define how an Event Cycle is to be calculated (i.e., specification of readers, notification latency, aggregates, filters). As a consequence, a standardized XML ECReport is generated and provided to the Capturing application.

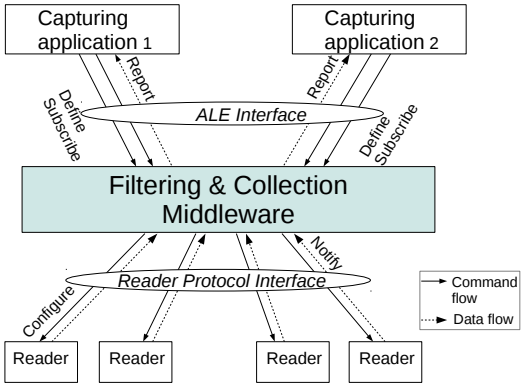

Fig. 3. EPCglobal Middleware

ALE 1.1 comprises five standards APIs, namely Reading, Writing, tag Memory, logical Reader and Access Control APIs. In this paper, we are interested in the Reading API which has the task to read tags and give reports in a variety of ways.

A. Reading API Context

To request for filtered data, specified by the ECSpec, ALE capturing applications can perform some actions (cf., Figure 3). These actions are denoted as: *Define/Undefine* an ECSpec to create a standing query, then to destroy it when no more needed. *Subscribe/Unsubscribe* to an ECSpec to subscribe a client to one of the event cycles already defined and to unsubscribe it when the client does no more need the service. *Poll/Immediate* are other methods of queries named "on-off queries", as they unsubscribe the client immediately after one event cycle. The ALE is also responsible for connecting readers to the middleware by defining/undefining logical readers and adding/removing them. These latter actions are not taken into account until the activation of the F&C middleware.

B. Limited Security Solution in the EPCglobal Specification

EPCglobal has proposed an API of access control to use the functionalities of the F&C middleware. This API allows administrative clients to define the access rights of other clients when using the methods and resources proposed by other ALE APIs (e.g., Reading, Writing APIs) in the F&C middleware. The model is role based access control. A role maps to one or more permissions. A permission describes an access to a particular feature of the ALE API. Two kinds of permissions exist: the *function permissions* and the *data permissions*. The first one grants the right to use a particular method of the ALE API (e.g. define, subscribe, unsubscribe in the Reading API), whereas the second one grants the right to use a particular resource or data (e.g., the right to govern a particular reader or to use another ALE API).

The security mechanism of the access control API limits the accesses to critical methods (i.e., subscription to capture tag data). Nevertheless, it does not cope with the details of data association and the use of filtered and combined reports within a request specification. We believe that a fine-grained security on these collected data for privacy concerns has to be handled to prevent applications, even in the same organization, to collect data that undermine people's private lives or reputations.

IV. Introducing the Privacy Control in the Filtering and Collection Middleware

Global System 1 has developed the EPC/RFID Privacy Impact Assessment (PIA) tool to help companies uncover the privacy risks and perform their own privacy assessments when implementing their RFID applications [17]. These privacy recommendations are defined in accordance with relevant privacy and data protection laws and regulations [10], [18]. Based on these directives, we summarize the main privacy statements in RFID environments in seven dimensions: (1) Purpose specification to announce the valid purpose associated with the collected data. (2) Collection limitation to ensure that only necessary information is collected. This should be done by lawful means or according to the data owner preferences. (3) Data owner access to correct, erase or amend his personal data. (4) Notification of the data owner to guarantee his right to be informed when new requests contains personal data. (5) Explicit consent of the data owner to ensure that the data is collected with the knowledge or consent of the data owner and prevent any unexpected use. (6) Security mechanisms designed to protect personal data from unauthorized access or disclosure (e.g., encryption, anonymity). (7) Collection relevance to ensure that data is up to date and retained with respect to the period of relevance.

TABLE I
Standard privacy dimensions in the middleware level

Dimension	Middleware support
(1) Purpose specification	with purpose declaration associated with the request
(2) Collection limitation	with accuracy and anonymity specifications in the request filter
(3) Data owner access	out of scope
(4) Notification	triggered for requests of new events (optional)
(5) Explicit consent	with storage of data owner preferences (e.g, temporal, spatial)
(6) Security mechanisms	by access control or encryption mechanisms
(7) Collection relevance and retention	out of scope

Some of these privacy dimensions can be expanded in the middleware level while others are dropped to suit its specifications. These latter could be rather enforced in upper layers of data processing. Table I shows the privacy requirements that can be achieved in the F&C middleware.

As data in the middleware stage are not designed to be stored (i.e., the case of Fosstrak platform), the specification of data owner access to amend data is not applicable. In addition, Collection relevance and retention cannot be applied as events are not stored in this stage.

A. Privacy Policy as a User-declared Context

There are several models of privacy in the literature. These models are used to integrate privacy requirements into a security policy [19], [20], [21], [22]. These security policies are generally specified according to an access control model to simplify the upgrade of existing information systems. Here, we detail the purpose dimension of the PrivOrBAC model [22], which extends the Organization-Based Access Control model (OrBAC) [23] by reusing most of its implemented existing mechanisms. The choice of PrivOrBAC relies on its capacity to handle most of the privacy requirements appearing in the guidelines and recommendations previously mentioned; but of course it can be substituted with another privacy model without disrupting the whole process.

OrBAC provides a set of concepts to express the security policy and enables making distinction between an abstract policy specifying organizational requirements and its concrete implementation in a given information system. Abstract organization privileges, such as *permission*, are expressed through the predicate *Permission(org, r, a, v, c)*. It means that the organization *org* grants a permission to role *r* to realize the activity *a* on the view *v* in the context *c*.

When declaring a context, a subject obtains some specific permissions and possibly also some obligations or prohibitions. For instance, a subject empowered in the role *nurse* may be permitted to declare that she is performing an *inspection*. By doing so, this subject will get the permission to have an access to the tag data that are in relation with her service.

We name as data owner, the individual who specifies a set of privacy preferences. These preferences depend on the context and the privacy dimensions supported by the application.

The purpose is modeled as a user-declared context. Each data owner can create purpose objects to specify the purposes for which access to his personal objects are allowed. The purpose objects, belonging to a finite set of *PO*, are grouped and inserted in a purpose view. Purpose values range over the purpose value domain *PV* (e.g., Medical_research, Inspection). Each purpose object has two attributes:

- *Recipient*, which is a predicate over domains *PO* × S defining who takes advantage of the declared purpose. That is, if *po* is a purpose object belonging to *PO* and *s* is a subject, then *Recipient(po,s)* means that *s* is the subject who takes advantage of the declared purpose *po*.
- *Declared_purpose*, which is a predicate over domains *PO* × *PV*, associating a purpose value with the declared purpose object. That is, if *po* is a purpose object and *pv* is a purpose value, then *Declared_purpose(po, pv)* means that *pv* is the purpose value associated with the declared purpose *po*.

By inserting a *po* in his purpose subview, a data owner declares that another subject (a *Recipient*) will perform some activity in a given context.

Let *do-purpose* denotes the data owner purpose subview associated with his defined objects. The purpose context is specified as follows: $\forall org \in Org, \forall s \in S, \forall \alpha \in A, \forall o \in O, \forall pv \in PV, \forall po \in PO,$
$Hold(org, s, \alpha, o, userdeclared(pv)) \leftarrow data\_owner(o, do)$
$\wedge Use(org, po, do\text{-}purpose) \wedge Recipient(po, s)$
$\wedge Declared\_purpose(po, pv)$[1]

That is, in organization org, subject s performs action α on object o associated with the data owner do, in the user declared context $user\_declared(pv)$, if there is a purpose object po used in the subview $do\text{-}purpose$ by organization org such that s is the recipient associated with po and pv is the declared purpose associated with po.

[1] $\wedge$: Logical conjunction

B. Use case: Monitoring system with user-declared context

Let us apply the defined model into a healthcare monitoring system using the F&C middleware.

The patient Bob is remotely monitored at home from different services of the Hospital H1. Bob is diabetic and also suffers from alzheimer. The medications he daily takes are tagged with EPC identification. The nurse of the diabetology service has to monitor Bob twice a day, i.e., every time he takes his diabetes medication, based on the tags read from his room reader. The nurse of the diabetology service has not to track the patient tags that are related to the alzheimer disease. This latter information is considered as private with respect to the nurse. We assume that (1) the nurse subscribes to an ECSpec (2) the defined ECSpec specifies the cycle time boundaries and the objects (tags) that the nurse wants to view, and (3) the tags are identified by their EPC code. Here, we are only interested in tags of type D. This means that the control is done over the field *product type* of the EPC code (cf., Figure 4) describing the reference of the diabetes medication.

Fig. 4. Tag of type D

1) The Rule:

- The $dataowner$ who is the patient or another person to whom the patient has delegated the right to control the access to his data defines the context $diabetes\_control$.
- In hospital $H1$, a user empowered in the role $nurse$ is permitted to have an access to the generated reports containing only tags of type D in the declared context $diabetes\_control$. The nurse has not to know other tagged objects.

2) The Modeled Rule: In the ALE middleware, the subscription request requires two subjects who are involved in the process of context declaration: the subject who is requesting the ALE middleware and the subject who takes advantage of this request. Our model considers the "real" recipient which is the *nurse*, independently of the one who has activated the subscription (i.e., it can be the patient nurse, or the doctor). In the following, we present our modeled rule:

- Step1: we first specify that the patient (or any empowered person if the patient is unable to do so) who is the $dataowner$ is permitted to define the purpose $diabetes\_control$ that applies to one of his nurses. This is done by the following permission rule:

  $Permission(H1, patient, declare, diabetes\_control, patient\_nurse)$,

  where $diabetes\_control$ is a purpose associated with some objects and $patient\_nurse$ is a context defined as follows:
  $\forall org \in Org, \forall s \in S, \forall \alpha \in A, \forall po \in PO,$
  (1) $Use(H1, po, diabetes\_control) \leftarrow Use(H1, po, do\text{-}purpose) \wedge Declared(po, diabetes\_control)$
  (2) $Hold(H1, s, \alpha, po, patient\_nurse)) \leftarrow Use(H1, po, do\text{-}purpose) \wedge PatienID(po, s) \wedge Recipient(po, s') \wedge Nurse(s', s)$

  That is, (1) the hospital $H1$ uses the purpose object po in the subview $do\text{-}purpose$ ($do\text{-}purpose$ is the $dataowner$ subview of the view $Purpose$ associated with his defined objects), such that $diabetes\_control$ is the declared purpose associated with po. In addition, (2) in hospital $H1$, a subject s performs an action α on the object o associated with the patient s in the context $patient\_nurse$ if there is a purpose object po used in the subview $do\text{-}purpose$ by $H1$ such that s' is the recipient associated with po (represented by the application-dependent predicate $Recipient(po, s')$) and s' is the nurse of the patient s (represented by the application-dependent predicate $Nurse(s', s)$).

- Step2: we specify that subjects empowered in the role $Nurse$ are permitted to consult objects belonging to the view $TagD\_type$ in the user-declared context $diabetes\_control$, as follows:

  $Permission(H1, Nurse, consult, TagD\_type, user\_declared(diabetes\_control))$

C. Privacy Controller Integration into the Fosstrak Platform

Relying on the presented interactions with the ALE middleware (cf., Section III) and the privacy dimensions required at this level (cf., Section IV), we consider that the subscription to an ECSpec is the action concerned with the privacy issues. As the defined ECSpec is not executed until the subscription method is performed, the collection of data is only performed by this action. The other activities in the middleware (cf., Section III-A) can be controlled by classical access control policies. In the sequel, we consider the subscription request to test our Privacy Controller module. Let's first present the Privacy Controller activity when triggered by a subscription request, then detail our testing scenario.

1) Privacy Controller Interactions and Related Algorithm:

To subscribe to an event cycle and receive related reports, the clients subscribe to an entity called ReportsGenerator [13] via the ALE interface. ReportsGenerator is the entity responsible for ensuring that the EventCycle is started/stopped and that subscribers receive the resulting reports. In this work, we define the PrivacyController as an entity responsible for verifying the compliance of the requests with a predefined privacy policy.

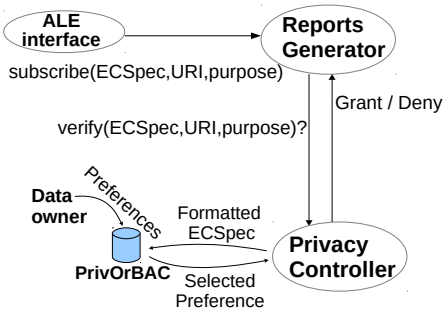

Fig. 5. Privacy controller interactions

Figure 5 depicts the relation between the two defined entities (i.e., ReportsGenerator and PrivacyController). The ReportsGenerator instantiates the PrivacyController with the specification ECSpec, the recipient address and the purpose to verify the properties of the request. This verification depends on the privacy policy and preferences defined by the data owner.

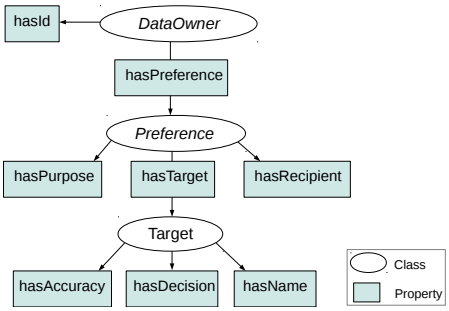

Fig. 6. Privacy preferences ontology

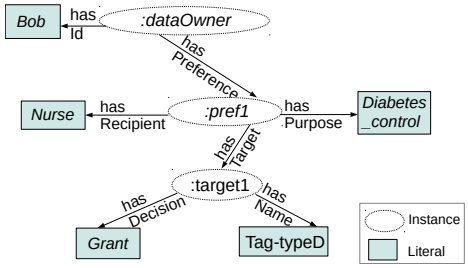

Fig. 7. Privacy preferences example

Preferences specification: The data owner specifies a set of privacy preferences depending on the privacy dimensions that are enforced in the system. We use the PrivOrBAC model [24] to administrate the privacy policy and the data owner preferences. As in Figure 6, each data owner $preference$ is associated with a $recipient$ address and a $purpose$ on which the preference is defined. This $preference$ turns around a set of targeted objects ($targets$) that the $recipient$ aims to access. A $decision$ attribute, related to each $target$, defines whether the data owner gives his consent to access the data or not and in the positive case, in which accuracy the data is disclosed.

Privacy control of subscriptions: Algorithm 1 shows the pseudo-code of the *subscribe* method included in the modified Fosstrak ReportsGenerator code. Steps (2,3) verify the existence of an already subscribed recipient, identified by its URI address with the same ECSpec. Steps (5,6) verify the permission of the recipient to subscribe to the reports notification with the entered ECSpec and purpose, relying on the defined privacy policy. Finally, Step 8 adds the address URI to the list of validated subscribers in case of permission.

In addition to the purpose and address of the recipient, the PrivacyController verifies the subscriber's targets in the ECSpec. If the subscriber specifies more than one $target$, the logical conjunction of the $decision$ attribute of all targeted objects is calculated as a final result. In our use case (cf., Section IV-B), the unique $target$ is the tag of type D. That is, according to the privacy policy, if the ECSpec contains only this $target$, a grant access to the data is returned. Then, the subscription successes. Figure 7 represents our studied use case where the patient Bob decides to disclose only the EPC code of his medications of type D in clear (without accuracy definition) for the declared purpose $diabetes\_control$.

Algorithm 1 Subscribe to ECSpec

Input: recipient `URI`, purpose `purp`, ECspec `spec`
1: NewSubscriber ← `URI`
2: **if** (Subscribers.containsKey(`URI`) **then**
3: ALEExceptionMessage ← "The `URI` is already subscribed to this specification `spec`"
4: **else**
5: **if** ! (PrivacyController.verify(`spec`, `URI`, `purpose`)) **then**
6: ALEExceptionMessage ← "The `URI` has not the right to subscribe to `ECspec` with the purpose `purp`"
7: **else**
8: Subscribers.put(`URI`)
9: **end if**
10: **end if**

Note that the purpose attribute, which corresponds to the declared user context, can either be added to the ECSpec XML content or to the *subscribe* request sent to the ALE middleware. We consider that adding this attribute to the *subscribe* request is more adequate for the following reasons. First, the ECSpec can be used by many ALE Clients and with possibly different purposes. Second, adding the purpose parameter to the request avoids distorting the ECSpec and leaves it consistent with the standard EPCglobal specification.

2) Testing Scenario & Results:

To test the subscription scenario, all roles and interfaces shown in Figure 3 have to be well configured, initialized and connected. We first give the implementation details that lead to our results. Then, we describe the messages flow related to the subscription activity applied into the F&C middleware.

a) Implementation details:

Our privacy module is added to the version 1.2.0 of the Fosstrak middleware. To connect the F&C middleware to different readers, we consider the EPCglobal Low Level Reader Protocol [25] (LLRP), as reader protocol via the LLRPCommander tool[2]. LLRPCommander provides a number of features to help configuring and managing LLRP-compliant RFID readers. To simulate LLRP readers and generate tag data, we use the reader emulator Rifidi[3]. As ALE Client (i.e., Capturing Application), we use the F&C Web Client application which comes with the Fosstrak platform. This Web Client is based on Java Server Pages (JSP). Finally, the PrivOrBAC policy is stored in a dedicated server and user's preferences are accessed via a web service [24], using the REpresentational State Transfer (REST) architecture.

b) Sequence diagram and generated report:

Figure 8 depicts a sequence diagram visualizing the messages flow related to the subscription activity with the new privacy controller module we have integrated into the Fosstrak platform. We assume that the client and the recipient of the report represent the same entity. The goal is to receive the reports corresponding to the Client specification, when the purpose and the recipient correspond to the permission rule defined in the privacy policy. The steps in Figure 8 are defined hereafter: 1: The Client subscribes to an EventCycle (via the ReportsGenerator), by entering the ECSpec (assumed to be created), the recipient address (URI) and the purpose parameters. Figure 9 illustrates a snapshot of an ECSpec specification rendered into XML. In this example, `0000389` refers to the manufacturer code for a given corporation, and `000162`

[2]http://code.google.com/p/fosstrak/wiki/LlrpMain
[3]http://www.transcends.co/community

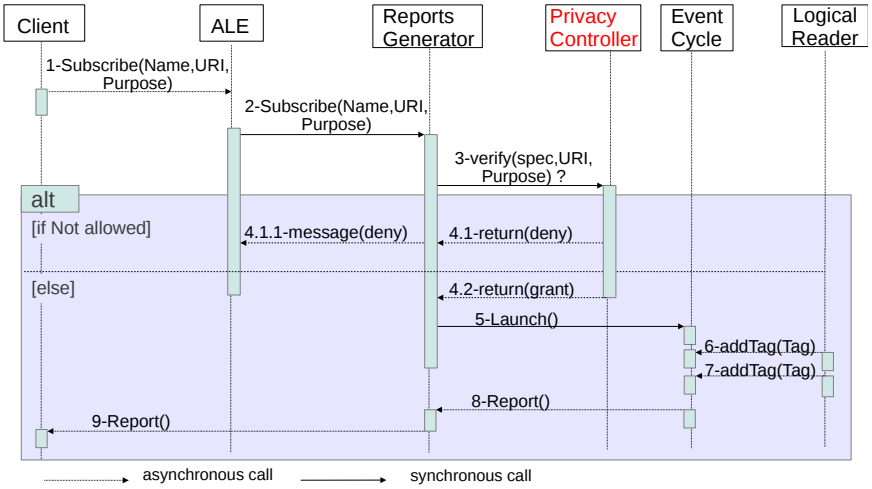

Fig. 8. Sequence diagram related to the subscription activity

denotes the product code for diabetes medication (Type D), where GTIN−96 tag encodings are used.

We assume that the nurse is only interested in the epc code, which matches the Type D tag, regardless of its serial number.

2: The ALE subscribes the Client to the ReportsGenerator that exclusively corresponds to this ECSpec.

```
<ns2:ECSpec>
...
 <reportSpecs>
  <reportSpec reportName="report1">
   <reportSet set="CURRENT"/>
    <filterSpec>
     <extension>
      <filterList>
       <filter>
        <includeExclude>INCLUDE</includeExclude>
        <fieldspec>
         <fieldname>epc</fieldname>
        </fieldspec>
        <patList>
         <pat>urn:epc:pat:sgtin-96:0000389.000162.*</pat>
        </patList>
       </filter>
      <filterList>
     <extension>
    <filterSpec>
    <output includeTag="true">
    </output>
   <reportSpec>
  </reportSpecs>
...
</ns2:ECSpec>
```

Fig. 9. Filtering in the ECSpec to obtain only epc tags of Type D

3: The ReportsGenerator instantiates the PrivacyController to verify the request depending on the privacy policy defined by the data owner. In our case, the control is based on each parameter of the ECSpec. Thus, the whole ECSpec (spec) is passed as parameter to the PrivacyController.

4.1/4.1.1: The PrivacyController checks the Client request. When the request does not meet the privacy policy (e.g., more than the Type D medication is specified or the URI Recipient or the purpose do not match the privacy preferences attributes of the data owner), a deny message is returned to the ALE interface stating that the privacy policy does not allow the present subscription.

4.2: The PrivacyController checks the request. In the present case, the access is granted, and therefore, the PrivacyController allows the ReportsGenerator related to the EventCycle to continue the processing.

5: The ReportsGenerator starts the EventCycle. The EventCycle now processes tags from the logical readers.

6/7: The LogicalReader(s) add(s) Tags to the EventCycle.

8/9: When the EventCycle reaches its boundaries, the reports are generated and sent back to the Client through the ReportsGenerator. Figure 10 shows a snapshot of the resulted ECReport presented in XML, reporting the set of tags read in the specified event cycle boundaries.

```
<ns2:ECReports>
...
 <report reportName="report1">
  <group>
   <groupList>
    <member>
     <tag>urn:epc:tag:sgtin-96:0000389.000162.1000</tag>
    </member>
    <member>
     <tag>urn:epc:tag:sgtin-96:0000389.000162.1001</tag>
    </member>
   </groupList>
  </group>
 </report>
</ns2:ECReports>
```

Fig. 10. Generated ECReport

V. Related Work

Most widely deployed middleware products do not fully consider the security requirements when processing sensitive data. Several implementations from software vendors [2], [3] and specialized companies [4] offering RFID middleware functions, manage the access control to historical events stored into final databases. However, they do not provide treatments to enhance privacy policies in accordance with international regulations, expected to be done at the middleware level.

Most existent implementations support EPCglobal network. Some of them explicitly include security solutions in the EPCIS level [2], [4], [3] relying mostly on role based access control and encryption solutions. Although some of these aforementioned products include security enhancements into their middleware implementations [2], [3], few authors worked on privacy problems. An exception is the work in [27], where the authors propose to support the EPCglobal middleware via a tool called *Privacy Framework Tool* plug-in. This tool includes privacy friendly practices and audits to be applied to the proposed middleware. To the best of our knowledge, there have

TABLE II
MIDDLEWARE SECURITY AND PRIVACY IN DIFFERENT PRODUCTS

	ALE version	Middl. Security	Middl. Privacy
IBM Websphere RFID [2]	1.1	Role based access control	-
Oracle sensor edge server [3]	1.0	Role Based user management	-
Fosstrak [13]	1.1	Authentication for ALE access	-
CHUK [26]	1.0	-	-
Aspire [27]	1.1	ALE Access Control API	-
WinRFID [4]	1.1	-	-

- : Not specified

been neither public communication related to the model used for privacy policy definitions nor information about the plug-in implementation. Table II summarizes some middleware security features of several platforms. It shows the version of the EPCglobal middleware interface (ALE) implemented in each tool and the proposed security and privacy approaches at this level. It is worth noting that the ALE 1.1 specification introduces several enhancements of the ALE 1.0, particularly a new API for controlling access to other ALE APIs.

So far, all the mentioned products propose a role based access control as a security approach in the middleware (i.e., ALE access control API is also based on this approach, as stated in Section III). However, these works still lack a privacy control integrated solution. For example, the authors do not cite the notion of declared user purpose to be applied in the definition of their policies. This notion is used in our proposition to extend the classical access control model and to enhance the policy by handling some privacy requirements in accordance with known regulations.

VI. CONCLUSION

In a context of assisting patients and elderly, collecting sensitive information about their health conditions is unavoidable. However, when associated with their identities, these information raises serious privacy concerns. In this paper, we have shown the importance of introducing privacy solutions at early stages of RFID data processing to prevent unintentional disclosures of sensitive information. The ultimate goal is to ensure that the communication line that extends from readers to middleware and enterprise applications responds to privacy requirements. First, we addressed some privacy concerns of the EPCglobal technology which is currently one of the predominant standardization efforts in the RFID community. Second, we provided a policy-driven approach to enforce privacy in such a technology. Third, we provided a proof-of-concept prototype that shows the feasibility of our approach with the purpose dimension. Some ongoing researches are related to the introduction of the accuracy dimension that defines different views depending on the recipient and its declared purpose.

REFERENCES

[1] M.-P. Schapranow, A. Zeier, and H. Plattner, "Security Extensions for Improving Data Security of Event Repositories in EPCglobal Networks," in *IFIP 9th International Conference on Embedded and Ubiquitous Computing (EUC)*, 2011, pp. 213–220.

[2] IBM Corp. (2010) IBM WebSphere Premises Server. [Online]. Available: http://www-01.ibm.com/software/integration/sensor-events/

[3] Oracle, "Oracle Application Server Wireless," Tech. Rep. 10.1.2, 2005. [Online]. Available: http://docs.oracle.com

[4] B. Prabhu, X. Su, H. Ramamurthy, C.-C. Chu, and R. Gadh, "WinRFID: A Middleware for the Enablement of Radiofrequency Identification (RFID)-Based Applications," *Mobile, wireless, and sensor networks: Technology, applications, and future directions*, p. 313, 2006.

[5] A. F. Westin, "Privacy and freedom," *Washington and Lee Law Review*, vol. 25, p. 166, 1968.

[6] S. Foresti, *Preserving Privacy in Data Outsourcing*, 1st ed. New York, NY, USA: Springer-Verlag New York, Inc., 2010.

[7] J. Biskup and J.-H. Lochner, "Enforcing confidentiality in relational databases by reducing inference control to access control," in *Information Security*. Springer, 2007, pp. 407–422.

[8] G. Aggarwal, M. Bawa, P. Ganesan, H. G-Molina, K. Kenthapadi, R. Motwani, U. Srivastava, D. Thomas, and Y. Xu, "Two can keep a secret: A distributed architecture for secure database services," *CIDR*, 2005.

[9] C. Floerkemeier, R. Schneider, and M. Langheinrich, "Scanning with a Purpose - Supporting the Fair Information Principles in RFID Protocols," in *UCS*, 2004, pp. 214–231.

[10] Commission of the European Communities, "Commission Recommendation on the implementation of privacy and data protection principles in Applications supported by radio-frequency identification," 2009.

[11] A. Appari, D. L. Anthony, and M. E. Johnson, "HIPAA Compliance: An Examination of Institutional and Market Forces (1,2)," 2009.

[12] W. Tounsi, J. Garcia-Alfaro, N. Cuppens-Boulahia, and F. Cuppens, "Securing the communications of home health care systems based on RFID sensor networks," in *CNSR*. IEEE, 2010, pp. 284–291.

[13] Fosstrak. (2009) Project License. [Online]. Available: http://fosstrak.googlecode.com/svn-history/r2112/legacy_website/license.html

[14] S. Sarma, D. L. Brock, and K. Ashton, "The networked physical world," Auto-ID Center, Tech. Rep. White Paper MIT-AUTOID-WH-001, 2000.

[15] C. Floerkemeier, C. Roduner, and M. Lampe, "RFID Application Development with the Accada Middleware Platform," *IEEE Systems Journal*, vol. 1, no. 2, pp. 82–94, 2007.

[16] EPCglobal. Inc, "The Application-Level Events (ALE) Specification, version 1.1 - Part 1: Core Specification," EPCGlobal, Tech. Rep., 2008.

[17] EPCglobal. Inc. (2011) Public Policy. [Online]. Available: http://www.gs1.org/epcglobal/public_policy

[18] O. J. of the European Communities, Ed., *Directive 95/46/EC of the European Parliament and of the Council on the protection of Individuals with regard to the processing of personal data and on the free movement of such data*, ser. 31, no. 281, 1995.

[19] Q. Ni, D. Lin, E. Bertino, and J. Lobo, "Privacy-Aware Role Based Access Control," in *12th ACM symposium on Access control models and technologies*. ACM, 2007, pp. 41–50.

[20] N. Yang, H. Barringer, and N. Zhang, "A purpose-based access control model," in *Third International Symposium on Information Assurance and Security (IAS)*. IEEE, 2007, pp. 143–148.

[21] A. Masoumzadeh and J. Joshi, "PuRBAC: Purpose-aware role-based access control," *On the Move to Meaningful Internet Systems (OTM)*, pp. 1104–1121, 2008.

[22] N. Ajam, N. Cuppens-Boulahia, and F. Cuppens, "Contextual privacy management in extended role based access control model," *Data Privacy Management and Autonomous Spontaneous Security*, pp. 121–135, 2010.

[23] A. A. E. Kalam, S. Benferhat, A. Miège, R. E. Baida, F. Cuppens, C. Saurel, P. Balbiani, Y. Deswarte, and G. Trouessin, "Organization based access control," in *POLICY*. 4th IEEE International Workshop on Policies for Distributed Systems and Networks, 2003.

[24] S. Oulmakhzoune, N. Cuppens-Boulahia, F. Cuppens, S. Morucci, M. Barhamgi, and D. Benslimane, "Privacy query rewriting algorithm instrumented by a privacy-aware access control model," in *Annals of telecommunications (ANTE)*, 2013.

[25] EPCGlobal Inc, "Low Level Reader Protocol (LLRP)," EPCGlobal, Tech. Rep. Version 1.1, 2010.

[26] Mobitec of CUHK IE department, "CUHK RFID System 1.0," Mobitec, Tech. Rep., 2007. [Online]. Available: http://mobitec.ie.cuhk.edu.hk/rfid/middleware/index.htm

[27] INRIA. (2009) ASPIRE-Advanced Sensors and lightweight Programmable middleware for Innovative RFID Enterprise applications. [Online]. Available: www.fp7-aspire.eu/