US 20080189758A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2008/0189758 A1

(43) Pub. Date: Aug. 7, 2008

Akeel et al.

(54) **PROVIDING SECURITY FOR QUERIES TO ELECTRONIC PRODUCT CODE INFORMATION SERVICES**

(75) Inventors: **Umair Akeel**, Burlingame, CA (US); **Steven P. Beier**, San Jose, CA (US); **Valer-Alin Crisan**, San Jose, CA (US); **Gautham B. Pai**, Hiriadka (IN); **Ralf Rantzau**, San Jose, CA (US)

Correspondence Address:
MOLLBORN PATENTS, INC.
ATTN: IBM
2840 COLBY DRIVE
BOULDER, CO 80305

(73) Assignee: **INTERNATIONAL BUSINESS MACHINES CORPORATION**, Armonk, NY (US)

(21) Appl. No.: **11/670,245**

(22) Filed: **Feb. 1, 2007**

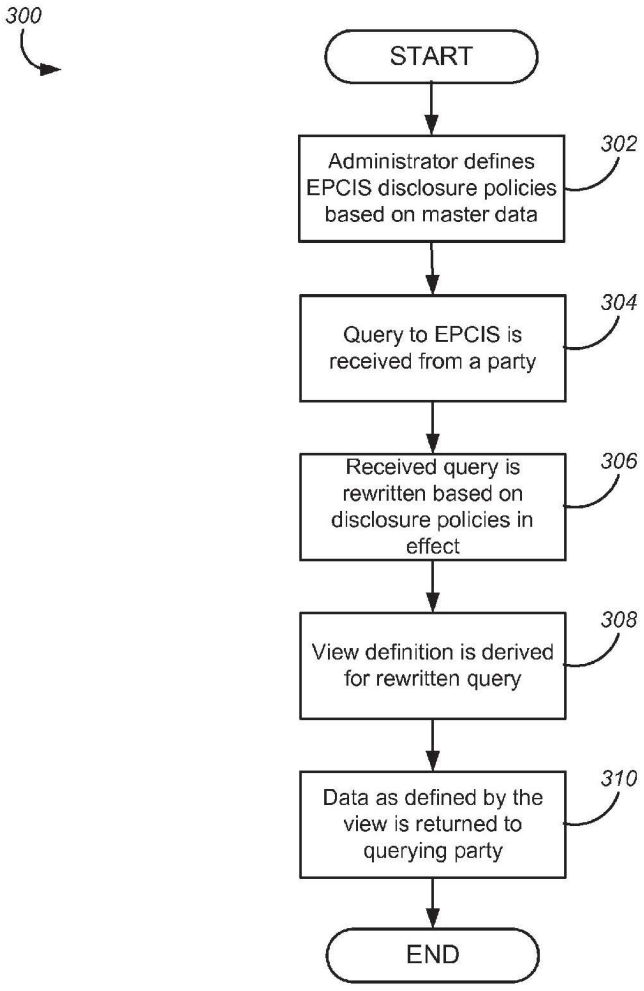
Publication Classification

(51) **Int. Cl.**
G06F 21/00 (2006.01)

(52) **U.S. Cl.** **726/1**

(57) **ABSTRACT**

Methods and apparatus, including computer program products, implementing and using techniques for protecting data residing in a repository of an electronic product code information service against undesired data disclosure. One or more disclosure policies are defined for an item tagged with an electronic product code. The item has associated data stored in the repository. The disclosure policies describe one or more of: who is permitted to query the repository for information, what type of information is permitted to be obtained from the repository in response to a query, and under what condition the repository can be queried. The disclosure policies are enforced in response to a received query from a party by only disclosing a subset of the data from the repository, the subset being determined in accordance with the defined one or more disclosure policies. A web-based tool for defining disclosure policies is also described.



OKTA, INC. 1047

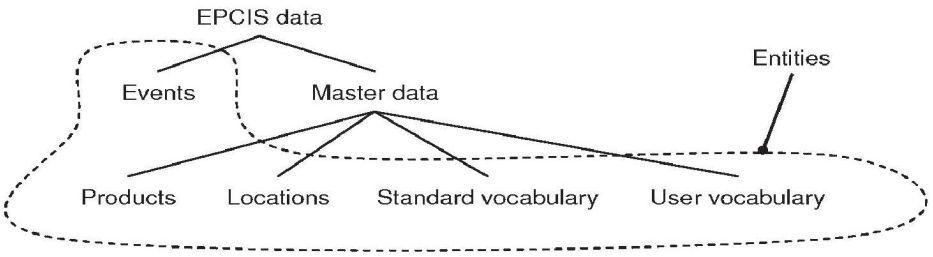

EPCIS data
Events
Master data
Entities
Products
Locations
Standard vocabulary
User vocabulary

FIG. 1

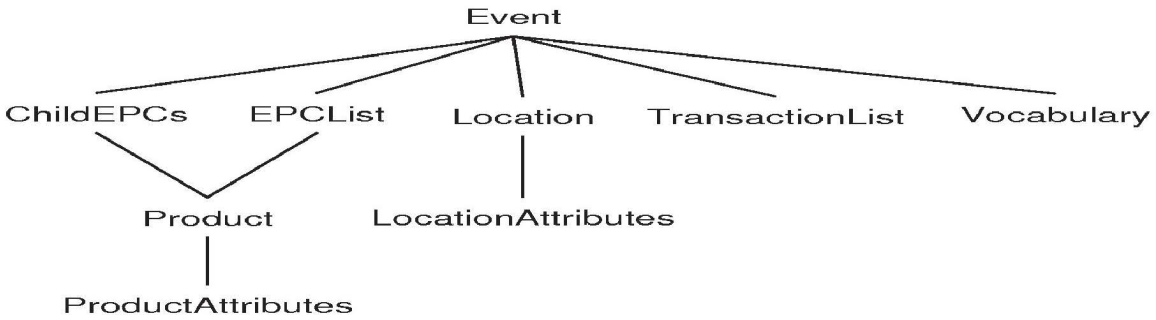

Event
ChildEPCs
EPCList
Location
TransactionList
Vocabulary
Product
LocationAttributes
ProductAttributes

FIG. 2

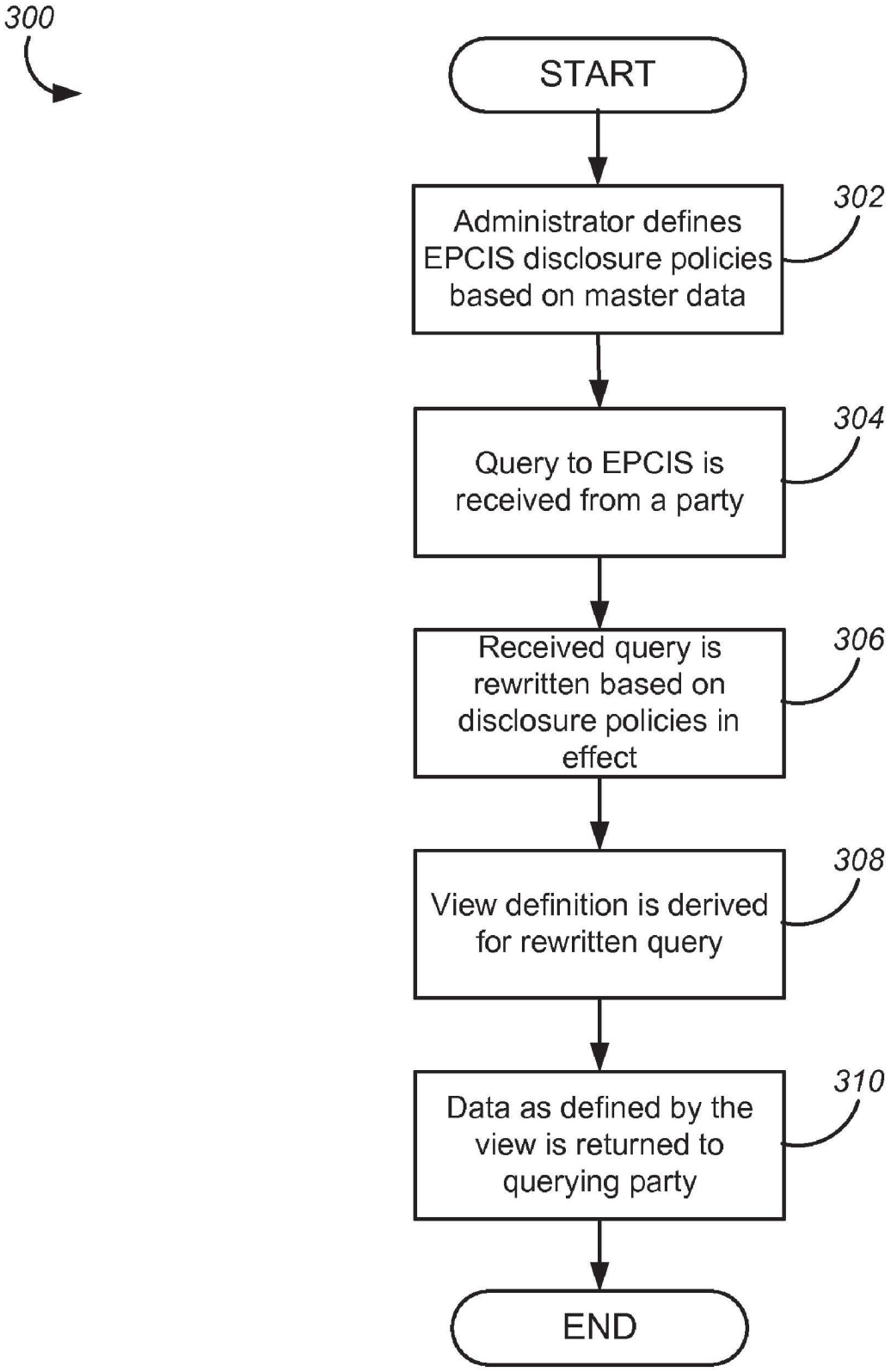

300
START
Administrator defines EPCIS disclosure policies based on master data
302
Query to EPCIS is received from a party
304
Received query is rewritten based on disclosure policies in effect
306
View definition is derived for rewritten query
308
Data as defined by the view is returned to querying party
310
END

FIG. 3


400
RFID Information Center
Security Policy Editor
Policy Information
Define the security policies and rules that control who can run queries and what data is disclosed to them.
406
Policies
Create
Delete
Import
Export
Enable
Disable
Toggle
404
Policy Name
Number Of Rules
User Group
State
Pharma supply chain -- Partner: Pharmacy 1
3
Pharmacy 1
Pharma supply chain -- Partner: Pharmacy 2
0
Pharmacy 2
Pharma supply chain -- Partner: Manufacturer 1
0
Manufacturer 1
Pharma supply chain -- Partner: Manufacturer 2
0
Manufacturer 2
Total policies : 4
402

FIG. 4

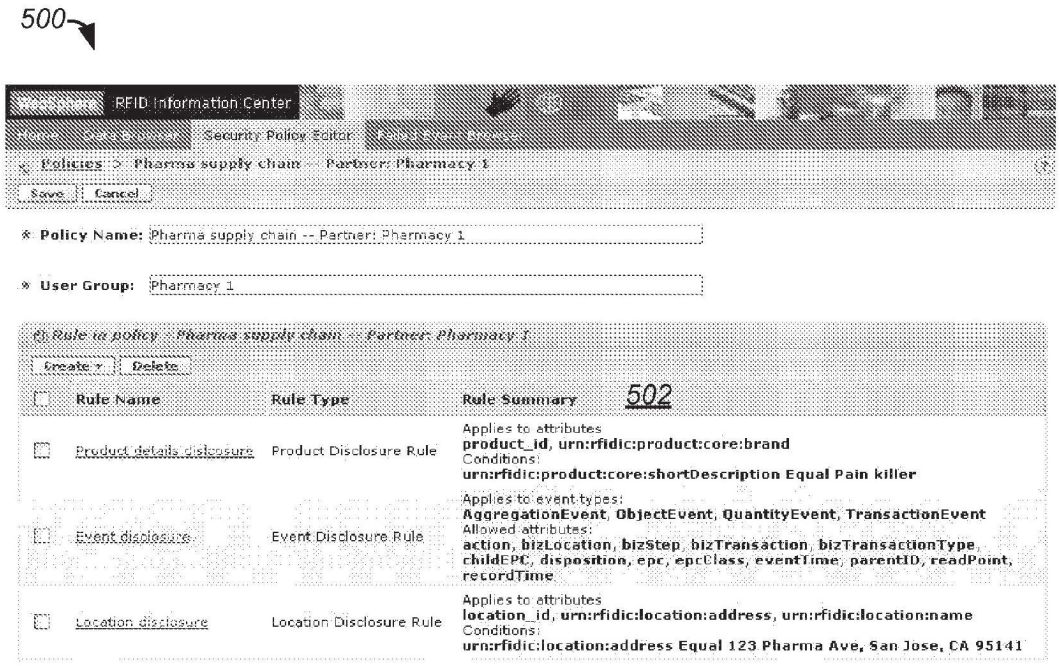

500
RFID Information Center
Security Policy Editor
Policies > Pharma supply chain -- Partner: Pharmacy 1
Save
Cancel
* Policy Name: Pharma supply chain -- Partner: Pharmacy 1
* User Group: Pharmacy 1
Rule to policy "Pharma supply chain -- Partner: Pharmacy 1"
Create
Delete
Rule Name
Rule Type
Rule Summary
502
Product details disclosure
Product Disclosure Rule
Applies to attributes
product_id, urn:rfidic:product:core:brand
Conditions:
urn:rfidic:product:core:shortDescription Equal Pain killer
Event disclosure
Event Disclosure Rule
Applies to event types:
AggregationEvent, ObjectEvent, QuantityEvent, TransactionEvent
Allowed attributes:
action, bizLocation, bizStep, bizTransaction, bizTransactionType, childEPC, disposition, epc, epcClass, eventTime, parentID, readPoint, recordTime
Location disclosure
Location Disclosure Rule
Applies to attributes
location_id, urn:rfidic:location:address, urn:rfidic:location:name
Conditions:
urn:rfidic:location:address Equal 123 Pharma Ave, San Jose, CA 95141

FIG. 5

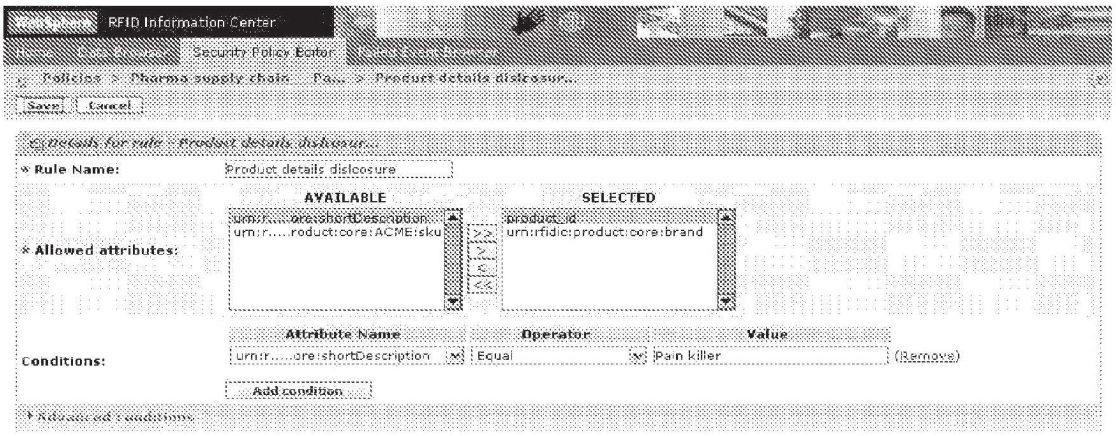

RFID Information Center
Security Policy Editor
Policies > Pharma supply chain - Pa... > Product details disclosu...
Save
Cancel
Details for rule - Product details disclosure
* Rule Name:
Product details disclosure
AVAILABLE
SELECTED
* Allowed attributes:
urn:r.....oreshortDescription
urn:r.....roduct:core:ACME:sku
product_id
urn:rfidic:product:core:brand
Attribute Name
Operator
Value
Conditions:
urn:r.....ore:shortDescription
Equal
Pain killer
(Remove)
Add condition
* Advanced conditions

FIG. 6

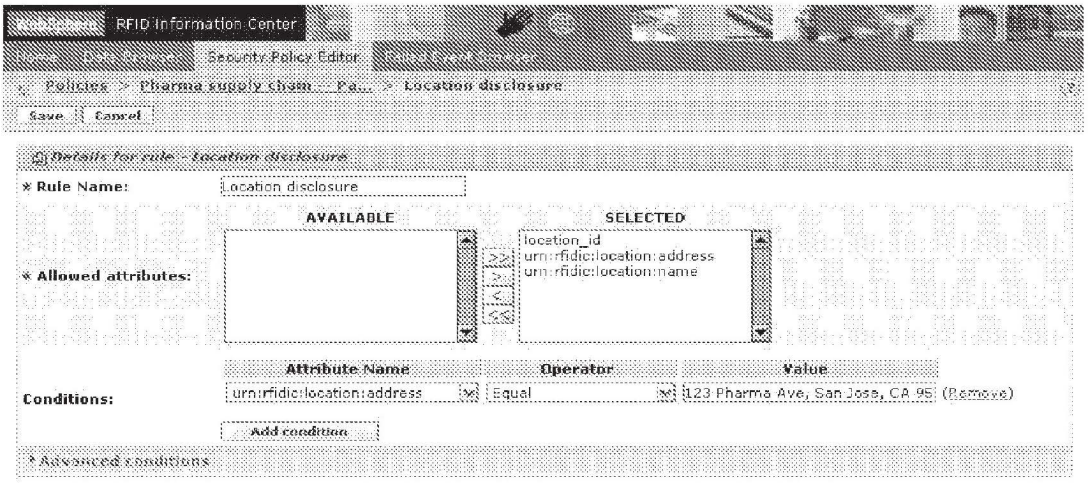

RFID Information Center
Security Policy Editor
Policies > Pharma supply chain - Pa... > Location disclosure
Save
Cancel
Details for rule - location disclosure
* Rule Name:
Location disclosure
AVAILABLE
SELECTED
* Allowed attributes:
location_id
urn:rfidic:location:address
urn:rfidic:location:name
Attribute Name
Operator
Value
Conditions:
urn:rfidic:location:address
Equal
123 Pharma Ave, San Jose, CA 95
(Remove)
Add condition
* Advanced conditions

FIG. 7

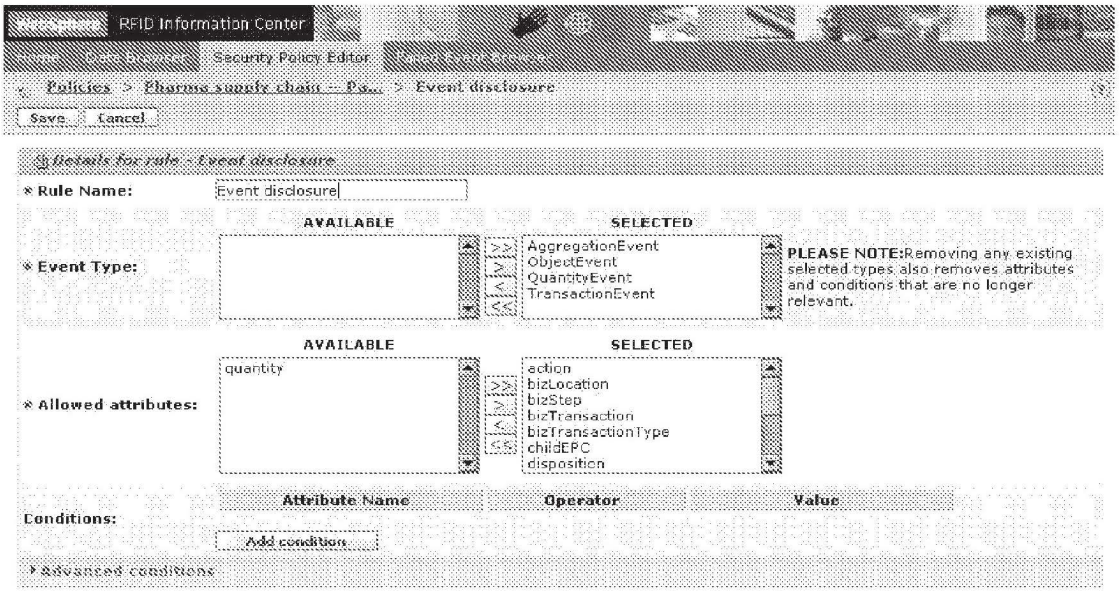

RFID Information Center
Security Policy Editor
Policies > Pharma supply chain -- Pa... > Event disclosure
Save
Cancel
Details for rule: Event disclosure
* Rule Name:
Event disclosure
AVAILABLE
SELECTED
* Event Type:
AggregationEvent
ObjectEvent
QuantityEvent
TransactionEvent
PLEASE NOTE:Removing any existing selected types also removes attributes and conditions that are no longer relevant.
AVAILABLE
SELECTED
* Allowed attributes:
quantity
action
bizLocation
bizStep
bizTransaction
bizTransactionType
childEPC
disposition
Attribute Name
Operator
Value
Conditions:
Add condition
Advanced conditions

FIG. 8

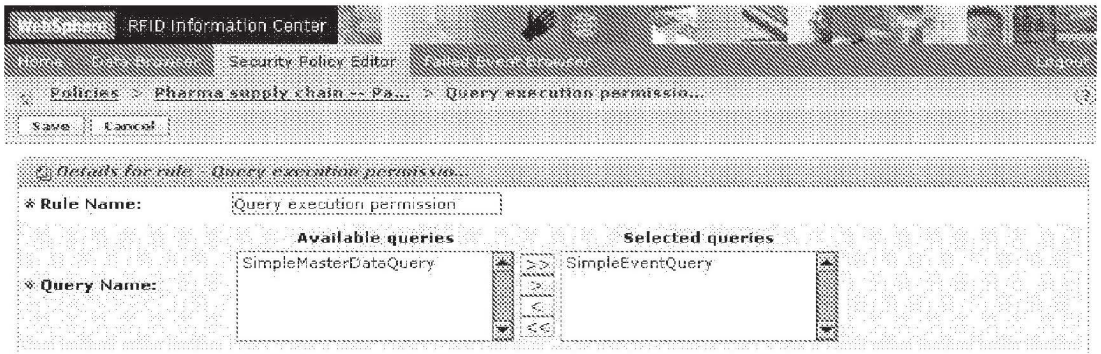

RFID Information Center
Security Policy Editor
Policies > Pharma supply chain -- Pa... > Query execution permissio...
Save
Cancel
Details for rule: Query execution permissio...
* Rule Name:
Query execution permission
Available queries
Selected queries
* Query Name:
SimpleMasterDataQuery
SimpleEventQuery

FIG. 9

PROVIDING SECURITY FOR QUERIES TO ELECTRONIC PRODUCT CODE INFORMATION SERVICES

BACKGROUND

[0001] This invention relates to security management of electronic product code information. Electronic Product Codes (EPC) is a family of coding schemes for RFID (Radio Frequency IDentification) tags. The EPC is designed to meet the needs of various industries, while guaranteeing the uniqueness for all EPC-compliant tags. All EPC numbers contain a header identifying the encoding scheme that has been used. This in turn dictates the length, type and structure of the EPC. EPC coding schemes typically contain a serial number which can be used to uniquely identify a single object. For example, a 96-bit EPC number allows approximately 1.3×10^{16} items to be coded annually, roughly corresponding to the number of grains of rice consumed globally each year.

[0002] The RFID and EPC technologies provide the basis for new traceability applications for items such as products to emerge everywhere. In this document, the term product refers to any physical object or item that is associated with an EPC. Parties that want to share their data in order to create increased business value need new mechanisms for querying traceable data. In addition to business considerations, companies are more likely to share their data if the services infrastructure is scalable, secure and easy to use.

[0003] The EPCglobal Network architecture describes components and interfaces for EPC-related information exchange between servers that contain information about items identified by EPC numbers. The servers, typically known as EPCISs (EPC Information Services), which are linked through a set of network services, store relevant product information related to specific EPC numbers. Typically, events are submitted directly from systems that sense EPCs, such as RFID readers and 1D- or 2D-barcode scanners. Events can also be submitted from systems that have received RFID data and "cleaned up" the data, for example, by applying filtering and transformation on the raw data events delivered by the RFID sensors. For example, a submitted event can describe a read operation of an RFID reader or an aggregation operation, where a set of EPCs is associated with another EPC. Such an aggregation operation can occur, for example, when several RFID-tagged items are packed into a container which is itself tagged with an RFID chip. The EPCISs are typically queried by other EPCIS systems, by extract-transform-load (ETL) systems that extract bulk data from the EPCIS and import the extracted data into a data warehouse for business intelligence applications, or by other customized applications that continuously monitor events.

[0004] The need for protecting information traces of items in supply chains is a challenge, in particular in new information infrastructures such as the EPCglobal networks, where partners want to share information, such as EPC events, with the help of EPCIS repositories. The current EPCglobal standard does not require EPCIS implementations to enforce authorization of queries, but suggests ideas of how an EPCIS could react in order to provide some notion of authorization. Some of the suggested reactions described in the standard document state that the EPCIS service could refuse to honor the request altogether, that the service could respond with less data than requested and that the service could hide information. However, no details are provided as to how these suggested reactions could be implemented in a secure way.

SUMMARY

[0005] In general, in one aspect, the invention provides methods and apparatus, including computer program products, implementing and using techniques for protecting data residing in a repository of an electronic product code information service against undesired data disclosure. One or more disclosure policies are defined for an item tagged with an electronic product code. The item has associated data stored in the repository. The disclosure policies describe one or more of: who is permitted to query the repository for information, what type of information is permitted to be obtained from the repository in response to a query, and under what condition the repository can be queried. The disclosure policies are enforced in response to a received query from a party by only disclosing a subset of the data from the repository, the subset being determined in accordance with the defined one or more disclosure policies.

[0006] In general, in another aspect, the invention provides methods and apparatus, including computer program products, implementing and using techniques for defining disclosure policies for data residing in a repository of an electronic product code information service. A user interface is presented through which a user can define one or more disclosure policies for the data residing in the repository. User input defining the one or more disclosure policies is received. The defined disclosure policies are applied to the data residing in the repository.

[0007] The various embodiments of the invention can realize one or more of the following advantages. Standard compliant implementations are provided that fulfill the suggestions about "query authorization" of the current EPCIS standard. As a result, users of EPCIS implementations do not need to be concerned about the security of their EPC-related information assets inside EPCIS repositories. By using master data to describe the conditions under which event data is disclosed, significantly less overhead is needed to ensure security for a potentially large number of events associated with a product. Data disclosure for entire product classes, that is, several individual items, can be controlled. In particular, the disclosure control can be based on specific product details such as brand, country of origin, or based on detailed location information about where the item was sensed, such as GPS coordinates, postal address, name of a distribution center, and so on.

[0008] The details of one or more embodiments of the invention are set forth in the accompanying drawings and the description below. Other features and advantages of the invention will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

[0009] FIG. **1** shows a schematic view of an entity in accordance with one embodiment of the invention.

[0010] FIG. **2** shows an excerpt of a data schema in accordance with one embodiment of the invention.

[0011] FIG. **3** shows a schematic process for enforcing data disclosure control in accordance with one embodiment of the invention.

[0012] FIG. **4** is a screenshot showing an overview of policies in accordance with one embodiment of the invention.

[0013] FIG. **5** is a screenshot showing an overview of the policy rules in accordance with one embodiment of the invention.

[0014] FIGS. **6-9** are screenshots showing policy rules in accordance with one embodiment of the invention.

[0015] Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

Overview

[0016] The need for protecting information traces of items in supply chains is a challenge, in particular in new information infrastructures such as the EPCglobal networks, where partners want to share information (typically EPC events) with the help of EPCIS repositories. The EPCIS query interfaces allow retrieval of events related to EPCs that have been submitted to the EPCISs' event capture interfaces, as well as information about the items that are associated with the EPCs.

[0017] These two kinds of data are referred to as event data and master data, respectively. Event data arises in the course of carrying out business processes, and is captured through an EPCIS Capture Interface and made available for query through the EPCIS Query Interfaces. Master data is additional data that describes a company's vocabulary, that is, the business entities, terminology, definitions and classifications used to describe business information, and that provides the necessary context for interpreting the event data. Informally, event data and master data can be understood as follows. Event data grows in quantity as more business is transacted, and refers to things that happen at specific moments in time. An example of event data is "At 1:23 pm on 15 Mar. 2004, EPC X was observed at Location L." Master data, on the other hand, does not generally grow merely because more business is transacted (though master data does tend to grow as organizations grow in size), is not typically tied to specific moments in time (though master data may change slowly over time), and provides interpretation for elements of event data. An example of master data is "Location L refers to the distribution center located at 123 Elm Street, Anytown, US." The rate of change to master data over time is generally expected to be very slow relative to the rate at which new event data is generated. Other examples of master data include locations of EPC sensing devices, such as the position of RFID readers inside a warehouse ("dock door 5" or "store floor" or "shelf G-16"), and product details such as color, size, or a best-before date for perishable goods.

[0018] The various embodiments of the invention described herein provide methods and apparatus that provide security and privacy for the query interface of EPCIS. A partner that queries an EPCIS can retrieve only those data values of an attribute in a query result object for which a policy rule has been specified for the attribute, and the condition of the policy rule is fulfilled for the result object. If no policy rule has been specified, the value is missing (or NULL in SQL terminology).

[0019] The various embodiments described herein provide methods for data disclosure enforcement for queries against EPCIS, which can use master data to control the data disclosure of query results, as well as tools for managing security policies that are used by the disclosure control methods. The methods include policy-based, role-based, fine-grained (cell-level) data disclosure control for EPC information, which encompasses both event information and master data, to protect unwanted disclosure of query results against EPCIS repositories. A data disclosure control technique is applied based on query rewriting. Queries that are submitted to an EPCIS through a query interface are intercepted transparently, that is, the querying partner does not have to know whether any security policies have been used to filter or suppress information stored by the EPCIS.

[0020] The tools for managing security policies, which will be described in detail below, have an intuitive graphical user interface that allows users to specify security policies for different user roles. The security policies pertain to which event attribute values and what master data attribute values can be disclosed and under what conditions. In some embodiments the tools allow one or more of creation, modification, deletion, activation and deactivation, and import and export of policies. The import and export functionality for policies is useful to easily deploy security policies in other EPCIS once the security policies have been defined for one EPCIS. In some embodiments the exported artifact is an XML file that has a well-defined document format.

[0021] In accordance with various embodiments of the invention, master data is used to describe the conditions under which event data is disclosed. As a result, the overhead needed to ensure security for a potentially large number of events associated with a product is greatly reduced. Typically, parties in a supply chain are not interested in ensuring the data visibility for a single EPC event or a single item (that is, an instance of a product), although, of course, this can also be done using the methods and apparatus in accordance with the various embodiments described herein. Instead, supply chain partners are typically interested in controlling data disclosure for an entire product class, that is, several items, and in particular based on specific product details, such as brand or country of origin, or based on detailed location information about where the item was sensed, such as GPS coordinates, postal address, name of a distribution center, and so on, which can all be described by master data.

[0022] As an illustrative example, consider a pharmaceutical wholesaler that manages its own EPCIS installation. The wholesaler has as partners M_1 and M_2, which are two competing pharmaceutical manufacturers that each delivers events to the wholesaler's EPCIS and query the EPCIS to track shipments. The wholesaler also has partners P_1 and P_2, which are two competing pharmacies. Pharmacy P_1 orders painkillers only from manufacturer M_1 and no products from manufacturer M_2. Pharmacy P_2 orders any type of products from both manufacturers M_1 and M_2. The EPCIS of the wholesaler receives EPC events about shipments from each partner. Each partner must also have access to their information in the wholesaler's EPCIS in order to manage their inventory and to track their shipments. To enable secure sharing of data, the EPCIS must ensure that M_1 is not able to query data pertaining to products of M_2 and vice versa, since they are competitors and want to protect their information assets. Furthermore, the EPCIS should allow each of P_1 and P_2 to see only shipment event information destined to them, respectively. P_1 should not be able to query for shipment and product detail information other than pain killers. How this is accomplished will now be explained in further detail with reference to the drawings. As the skilled reader realizes, the description is not limited to the specific examples described herein, but

should rather be viewed as explanatory examples of embodiments of the invention, as defined in the claims.

Entities

[0023] In some embodiments of the invention, the objects referenced in policy rules are defined on a higher conceptual level than objects of the EPCIS repository, thus shielding the user from implementation details when using the graphical user interface of the policy-definition tool, as will be described below. These high-level objects, referenced to by the policies, are referred to herein as entities. The entity concept is schematically illustrated in FIG. **1**. As can be seen in FIG. **1**, the EPCIS standard describes two types of data: events and master data. The master data, in turn, can include different types of context information, such as products, locations and vocabularies. Example attributes of products are the brand name or a stock keeping unit (SKU) identifier. Locations can have attributes like address, read point, or business location name. A vocabulary is the set of permissible values that can appear in an event or master data attribute, such as a set of available location names. Vocabularies typically provide context and define standard terms for business steps, dispositions, and transaction types. It should be noted that the term "entity" is not used herein as a database term (that is, it does not mean "entity" according to the entity-relationship (ER) model). An entity is any one of: event, location, product, further context that may be specified in a vocabulary.

[0024] As a result of providing an entity abstraction of the information specified in a physical data model, a person defining security policies does not need to have the skills or knowledge of a database administrator. For example, the person defining the security policies does not need to know that EPC values are stored in an EPCList table but can only see that an EPC is one among several attributes of an entity named event, or that an address is an attribute of an entity named location, or that brand is an attribute of the an entity named product, as will be discussed in further detail below with reference to FIG. **5**. The linking of events, products, locations, and further context information with each other and the linking of all lower-level objects that belong to an entity is done transparently to the user issuing the query.

[0025] The structure of the physical data model, including the relationships between entities, is defined in the metadata. Additionally, metadata allows the administrator of an EPCIS to define extensions to the EPCIS standard schema for events and master data, such as new attributes for events or for products. In some embodiments of EPCISs, metadata is defined in the form of XML documents that conform to defined XML schemas and that are used by a software component to create (deploy) the physical data model. This lower-level data model consists of several relational database objects for each entity as well as several database relations for security policies.

Physical Data Model

[0026] The user who sets up security policies does not need to be aware that the entities in the EPCIS repository are represented using a different data model, such as a relational database schema. A small excerpt of the schema used in one implementation of EPCIS is schematically illustrated in FIG. **2**. As can be seen in FIG. **2**, in the illustrated implementation, the events have a relationship with location details. In fact, according to the EPCIS standard, there are two relationships; a first relationship for the business location and a second relationship for the RFID reader location, but only one relationship type is shown in FIG. **2**, for the sake of clarity. The location attributes can be represented by a separate LocationsAttributes table.

Query Execution Control

[0027] FIG. **3** shows a process (**300**) for enforcing data disclosure control in accordance with one embodiment. As can be seen in FIG. **3**, the process starts with an EPCIS administrator defining a disclosure policy for the EPCIS (step **302**). The disclosure policy includes a set of one or more rules, typically one rule for each type of entity. It should be noted that not every entity must have a rule defined. Depending on the metadata definitions, a rule for one entity can refer to several database objects (typically, tables and views). For example, if a product policy rule declares the attributes that can be disclosed and the conditions under which these attributes can be disclosed, both the Product and the ProductAttributes tables are affected, as can be seen in FIG. **2**. In some implementations, the effect of defining policies on entities is that security views on relational database objects are created. If complex operators are used in the query interface such as the hypothetical EPCIS query operator "get-children-of" for traversing a hierarchy, the resulting view definitions may be (deeply) nested and may reference several database objects.

[0028] Next, a query is submitted to the EPCIS (step **304**). The query is rewritten based on the policies that are in effect (step **306**). An incoming query is first translated such that the query references the database objects according to the relationships between database objects defined in the metadata. Then, the query is transformed such that the query uses the appropriate security views instead of the original database objects.

[0029] After rewriting the query, view definitions are derived from the policy rules (step **308**). By way of example, the derivations of the view definitions will be described here in terms of relational database objects and a relational database system. It should however be noted that the derivation of the view definitions can also be implemented using a different data model. In the relational database implementation of the EPCIS repository, attributes used in queries map to columns in database tables or views.

[0030] For each combination of a user role r and a table t in a database that is to be protected according to policies and the metadata, a special view named v_r_t exists that has the same schema as t and that defines which table cells in t that can be disclosed to r. In most real-world application domains for EPC data repositories, the number of roles |r| and the number of tables |t| are small and the maximum number of views is |r|*|t| is sufficiently small for all practical purposes.

[0031] The query rewrite method is based on a table semantics model, where the main idea is that attribute values which are not to be disclosed will be replaced by NULL values. For each combination of a table t that can be queried according to the query interface of the EPCIS and a user role r, zero or more security policies can be created. For practical purposes, the number of security policies can be restricted to zero or one policy per role and table. Each policy is a collection of rules. A rule specifies the name of the table (t), the set of column names of t whose values may be disclosed (A), and the condition (c), for example, a complex Boolean expression, under which the values in A for a row in t may be disclosed.

[0032] A policy rule can be compared to an SQL view on table t, where A is the SELECT list, and c is the expression in the WHERE clause, illustrated below.

```
CREATE VIEW
<view_over_t_for_role_r_according_to_the_policy_rule>
    AS
SELECT A
FROM t
WHERE c
```

[0033] Finally, the data defined by the view is returned to the querying party (step **310**). The data returned by the view defines exactly the attribute values of rows in t that may be disclosed to users having the role r. No other values of rows in t will be disclosed.

[0034] When a query q that references a table t is issued by a user who has role r, every occurrence of a table tin q is replaced by its respective view v_r_t, as will be described in further detail below by way of example. These views are created when the security polices for r and t are created. If any policy for the combination r and t is modified, the view v_r_t is dropped and a new view definition with the same name is created. The new view again reflects the current disclosure rules.

[0035] In one embodiment, the view definition method works as follows. Let P be the set of all policies. Let (r, t, A, c) be a policy rule. Let $U(r', t')=\{(r, t, A, c)\in P | r=r' \wedge t=t'\}$ be the set of rules for a table t' of policies that pertain to a role r'. The WHERE clause of the view is a logical OR of all conditions in U. For each column name a of t, if the column name does not appear in an attribute set A in U, the column appears in the column list of the SELECT clause as:

[0036] NULL as a

[0037] Otherwise, let C be the set of conditions that have been defined for attribute a: $C(r', t', a)=\{p.c | p\in U(r', t') \wedge a\in p.A\}$. Then, the column a appears as an expression that involves a logical OR of all conditions c1, . . . , cn in C:

[0038] CASE WHEN <c1> OR . . . OR <cn> THEN a ELSE NULL AS a

Enabled and Disabled Policies

[0039] In various embodiments of the invention, a security administrator of an EPCIS can activate/enable or deactivate/disable a subset of the policies. This can be convenient when an EPCIS installation is undergoing security tests by a security administrator. Only the policies that are enabled are used in the security view definition process. Enabling and disabling individual policies can be implemented by dropping all policies and creating new views according to the current policies. Of course, it is also possible to analyze what policies have been disabled and enabled and to drop and recreate only those views whose definition needs to change while leaving all other views unchanged.

A Security Policy Example

[0040] The rewriting of queries will now be explained in greater detail by way of example of views that are generated based on policies and how these views are used to rewrite a query. First, assume there is a hypothetical entity E (a, b, c, d, e, f g). The metadata declares that E refers to two relational tables t_1 (a, b, c, d, e) and t_2 (a,f,g) (primary keys underlined), where $t_2.a \rightarrow t_1$ is a foreign key constraint between the two tables.

[0041] The user has been assigned to role r and three policies p_1, p_2, and p_3 have been created that are all enabled. The policies are illustrated below in Table 1.

TABLE 1

Policy p_1 for role r	Policy p_2 for role r	Policy p_3 for role r
Rule 1: Table: t_1 Attributes: a, b Condition: c_1	Rule 1: Table: t_1 Attributes: a, c Condition: c_2	Rule 1: Table: t_1 Attributes: a, c, d Condition: c_3 Rule 2: Table: t_2 Attributes: a, g Condition: c_4

[0042] The resulting views $v_r_t_1$ and $v_r_t_2$ formulated as SQL DDL statements, where $<c_i>$ are placeholders for the conditions, are:

```
CREATE VIEW v_r_t1 AS
SELECT a                                                AS a,
    CASE WHEN <c1>              THEN b ELSE NULL        AS b,
    CASE WHEN <c2> OR <c3>      THEN c ELSE NULL        AS c,
    CASE WHEN <c3>              THEN d ELSE NULL        AS d,
    NULL                                                AS e
FROM t1
WHERE <c1> OR <c2> OR <c3>
CREATE VIEW v_r_t2 AS
SELECT  a                                       AS a,
        NULL                                    AS f,
        g                                       AS g
FROM    t2
WHERE   <c4>
```

[0043] It should be noted that the first element in the SELECT list of v_r_t1,

[0044] a AS a

[0045] could have been written as

[0046] CASE WHEN <c2> OR <c2> OR <c3> THEN a ELSE NULL AS a

without altering the semantics of the query. This optimization can advantageously be incorporated into various embodiments. The reason is that the WHERE clause of the query will select only rows that fulfill the expression $c1 \vee c2 \vee c3$. Similarly, the third element of the select list of view v_r_t2,

[0047] g AS g

is equivalent to

[0048] CASE WHEN <c4> THEN g ELSE NULL AS g

[0049] Based on these policies, suppose a user in role r submits the query "Get the attribute values for b, c, f, and g from all objects of entity E," which can be translated into the following SQL query:

```
SELECT  b, c, f, g
FROM    t1, t2
WHERE   t1.a = t2.a AND
        <c5>
```

[0050] The rewrite method produces the following query by replacing the table references by view names and adjusting the attribute names to reference the respective view.

```
SELECT   b, c, f, g
FROM     v__r__t1, v__r__t2
WHERE    v__r__t1.a = v__r__t2.a AND
         <c5>
```

Security Policy Editor Tool

[0051] As was described above, various embodiments provide a web-based graphical user interface tool, referred to herein as a "Security Policy Editor," which can be used to manage the disclosure rules for queries against an EPCIS. This Security Policy Editor will now be described in further detail with reference to the above example of an EPCIS of a pharmaceutical wholesaler that has several trading partners that need to query the EPCIS repository in a secure way. The main focus of the following description will be on the policy rules for one of the partners, Pharmacy **1**.

[0052] FIG. **4** is a screenshot (**400**) showing an overview of the policies that are defined for the wholesaler's EPCIS. As can be seen in FIG. **4**, every partner has their own policy that has been set up by the security administrator of the EPCIS. Three policy rules have been defined for Pharmacy **1**. The state indicators (**402**) show that all policies are enabled, that is, queries coming from any of the partners (that is, Manufacturer **1**, Manufacturer **2**, Pharmacy **1** and Pharmacy **2**, respectively) are under disclosure control. The checkboxes (**404**) indicate that each policy can be selected and subject to one or more operations that are defined by the clickable buttons (**406**) above the policy listing. That is, each policy can be exported, enabled or disabled, a new policy can be imported, and the states of a set of selected policies can be toggled. Toggling means that if a policy is selected in a checkbox, its state changes, that is, it either changes from enabled to disabled or from disabled to enabled.

[0053] FIG. **5** shows an overview (**500**) of the three policy rules pertaining to Pharmacy **1**. As can be seen in FIG. **5**, the rule summary (**502**) gives the security administrator of the EPCIS a concise description of the disclosed attributes and the conditions under which the attributes may be disclosed. The policy name and user group are also clearly displayed, and options for creating and deleting policies are provided.

[0054] FIG. **6** shows details for the first rule belonging to the policy of the partner "Pharmacy **1**" that declares that any location information (including associated events) that pertains to products which are labeled "Pain killer" can be disclosed to Pharmacy **1**. Furthermore, only the product identifier and the brand name are disclosed as product details; the product details "short description" and SKU would yield the value NULL in queries that reference these attributes.

[0055] FIG. **7** shows details for the second rule belonging to the policy of the partner "Pharmacy **1**" that declares that any information (including events) that pertains to events whose origin is the location "123 Pharma Ave, San Jose, Calif. 95141" can be disclosed to Pharmacy **1**. No attributes of the location details are suppressed.

[0056] FIG. **8** shows details for the third rule belonging to the policy of the partner "Pharmacy **1**." The effect of this rule is that queries coming from Pharmacy **1** that involve event information will never disclose quantity information. If this attribute is referenced in a query of Pharmacy **1**, its values will be NULL.

[0057] Because of the policy rules for Pharmacy **1**, shown in FIGS. **5-7**, any query about EPC events against the EPCIS that comes from Pharmacy **1** will return only information that pertain to pain killers only and the location of the events that originate from the location "123 Pharma Ave, San Jose, Calif. 95141." Typically, the location describes where the (RFID, 1D- or 2D-barcode, and so on) readers have sensed the EPC attached to an item. However, events can also pertain to activities other than "reading" codes. An example is packing and unpacking of goods, which is expressed by the so-called AggregationEvent type, defined by EPCglobal (see the event type values in FIG. **8**).

[0058] FIG. **9** shows another rule belonging to the policy of "Pharmacy **1**." It describes what types of queries (for example, as a web service) Pharmacy **1** can issue to the EPCIS. As can be seen in FIG. **9**, only queries of type SimpleEventQuery can be executed, but not SimpleMasterData queries. These types of queries are defined in the EPCglobal standard for EPCIS.

[0059] The invention can take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment containing both hardware and software elements. In a preferred embodiment, the invention is implemented in software, which includes but is not limited to firmware, resident software, microcode, etc.

[0060] Furthermore, the invention can take the form of a computer program product accessible from a computer-usable or computer-readable medium providing program code for use by or in connection with a computer or any instruction execution system. For the purposes of this description, a computer-usable or computer readable medium can be any apparatus that can contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device.

[0061] The medium can be an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system (or apparatus or device) or a propagation medium. Examples of a computer-readable medium include a semiconductor or solid state memory, magnetic tape, a removable computer diskette, a random access memory (RAM), a read-only memory (ROM), a rigid magnetic disk and an optical disk. Current examples of optical disks include compact disk-read only memory (CD-ROM), compact disk-read/write (CD-R/W) and DVD.

[0062] A data processing system suitable for storing and/or executing program code will include at least one processor coupled directly or indirectly to memory elements through a system bus. The memory elements can include local memory employed during actual execution of the program code, bulk storage, and cache memories which provide temporary storage of at least some program code in order to reduce the number of times code must be retrieved from bulk storage during execution.

[0063] Input/output or I/O devices (including but not limited to keyboards, displays, pointing devices, etc.) can be coupled to the system either directly or through intervening I/O controllers.

[0064] Network adapters may also be coupled to the system to enable the data processing system to become coupled to other data processing systems or remote printers or storage devices through intervening private or public networks.

Modems, cable modem and Ethernet cards are just a few of the currently available types of network adapters.

[0065] A number of implementations of the invention have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. For example, the query interface of the EPCIS may be realized as a web service or an application programming interface (API). Entities other than events, products, and locations may be used to specify security policies. A user vocabulary typically specifies such further context types. Furthermore, although the EPCglobal consortium specifies standards for products, items that may be tagged could also be human beings, money, and parts of products (not full products). It should be clear that EPCIS can be used also for tracking these other types of objects. Accordingly, other embodiments are within the scope of the following claims.

1. A computer-implemented method for protecting data residing in a repository of an electronic product code information service against undesired data disclosure, the method comprising:

defining one or more disclosure policies for an item tagged with an electronic product code, the item having associated data stored in the repository, the one or more disclosure policies describing one or more of: who is permitted to query the repository for information, what type of information is permitted to be obtained from the repository in response to a query, and under what condition the repository can be queried; and

enforcing the one or more disclosure policies in response to a received query from a party by only disclosing a subset of the data from the repository, the subset being determined in accordance with the defined one or more disclosure policies.

2. The method of claim **1**, wherein the one or more disclosure policies are based on master data for the item, the master data describing one or more aspects of a business vocabulary for the item, including one or more of: business entities, terminology, definitions and classifications used to describe business information.

3. The method of claim **1**, wherein enforcing the one or more disclosure policies includes refusing to respond to the received query.

4. The method of claim **1**, wherein enforcing the one or more disclosure policies includes responding to the received query with less information than specified in the query.

5. The method of claim **1**, wherein defining one or more disclosure policies includes:

defining a role to be associated with one or more querying parties; and

associating one or more disclosure policies with the role.

6. The method of claim **1**, wherein defining one or more disclosure policies includes:

defining the one or more disclosure policies on an entity object level, wherein each entity object includes a predefined mapping to data attributes in the repository.

7. The method of claim **1**, wherein enforcing the one or more disclosure policies is done transparently without the querying party being aware that the one or more disclosure policies is applied to the query.

8. The method of claim **1**, further comprising:

receiving a query from a party, the query pertaining to data associated with an item and stored in the repository;

rewriting the received query to create a database view of the repository that is in accordance with the defined one or more disclosure policies; and

returning data defined by the database view to the querying party.

9. The method of claim **1**, further comprising:

providing an option for a disclosure policy administrator to selectively enable or disable individual disclosure policies.

10. A computer program product for protecting data residing in a repository of an electronic product code information service against undesired data disclosure, comprising a computer useable medium including a computer readable program, wherein the computer readable program when executed on a computer causes the computer to:

define one or more disclosure policies for an item tagged with an electronic product code, the item having associated data stored in the repository, the one or more disclosure policies describing one or more of: who is permitted to query the repository for information, what type of information is permitted to be obtained from the repository in response to a query, and under what condition the repository can be queried; and

enforce the one or more disclosure policies in response to a received query from a party by only disclosing a subset of the data from the repository, the subset being determined in accordance with the defined one or more disclosure policies.

11. The computer program product of claim **10**, wherein the one or more disclosure policies are based on master data for the item, the master data describing one or more aspects of a business vocabulary for the item, including one or more of: business entities, terminology, definitions and classifications used to describe business information.

12. The computer program product of claim **10**, wherein the computer readable program when executed on a computer further causes the computer to:

receive a query from a party, the query pertaining to data associated with an item and stored in the repository;

rewrite the received query to create a database view of the repository that is in accordance with the defined one or more disclosure policies; and

return data defined by the database view to the querying party.

13. A computer program product for defining disclosure policies for data residing in a repository of an electronic product code information service, comprising a computer useable medium including a computer readable program, wherein the computer readable program when executed on a computer causes the computer to:

present a user interface through which a user can define one or more disclosure policies for the data residing in the repository;

receive user input defining the one or more disclosure policies; and

apply the defined disclosure policies to the data residing in the repository.

14. The computer program product of claim **13**, wherein the user input defining the one or more disclosure policies includes user input describing one or more of: who is permitted to query the repository for information, what type of information is permitted to be obtained from the repository in response to a query, and under what condition the repository can be queried.

15. The computer program product of claim **13**, wherein the one or more disclosure policies are based on master data, the master data describing one or more aspects of a business vocabulary for the item, including one or more of: business entities, terminology, definitions and classifications used to describe business information.

16. The computer program product of claim **13**, wherein the user input includes a definition of an action to be taken in response to a query, the action including one or more of: responding with all the requested information, responding with less information than requested, and not responding to the query.

17. The computer program product of claim **13**, wherein the user input includes:

a definition of a role to be associated with one or more querying parties; and

an association of one or more disclosure policies with the role.

18. The computer program product of claim **13**, further comprising instructions causing the computer to:

receive user input selectively enabling or disabling individual disclosure policies.

19. The computer program product of claim **13**, wherein the one or more disclosure policies are defined on an entity object level, wherein each entity object includes a pre-defined mapping to data attributes in the repository.

20. The computer program product of claim **13**, wherein the user interface is a web-based user interface.

* * * * *