Distrib Parallel Databases (2011) 29:397–443
DOI 10.1007/s10619-011-7084-9

RFID enabled traceability networks: a survey

Yanbo Wu · Damith C. Ranasinghe · Quan Z. Sheng · Sherali Zeadally · Jian Yu

Published online: 4 June 2011

Abstract The emergence of radio frequency identification (RFID) technology brings significant social and economic benefits. As a non line of sight technology, RFID provides an effective way to record movements of objects within a networked RFID system formed by a set of distributed and collaborating parties. A trail of such recorded movements is the foundation for enabling traceability applications. While traceability is a critical aspect of majority of RFID applications, realizing traceability for these applications brings many fundamental research and development issues. In this paper, we assess the requirements for developing traceability applications that use networked RFID technology at their core. We propose a set of criteria for analyzing and comparing the current existing techniques including system architectures and data models. We also outline some research opportunities in the design and development of traceability applications.

Keywords RFID · Traceability · Networked RFID · Internet-of-Things · Data model · Traceability queries

1 Introduction

Traceability refers to the capability of an application to track the state (e.g., location, temperature) of goods, discover information regarding its past state and potentially

Communicated by Elisa Bertino.

Y. Wu (✉) · D.C. Ranasinghe · Q.Z. Sheng · J. Yu
The University of Adelaide, Adelaide, SA 5005, Australia
e-mail: yanbo.wu@adelaide.edu.au

S. Zeadally
Department of Computer Science and Information Technology, University of the District of Columbia, Washington, DC, USA

J. Yu
Swinburne University of Technology, John Street, Hawthorn, Melbourne, Victoria 3122, Australia

estimate its future state. Traceability is vital for efficient business operations and for making effective decisions, which is fundamental to a wide range of business applications such as inventory control, distribution planning, manufacturing control, product recalls, counterfeit detection and re-usable asset management.

Effective and accurate identification is very important to realize a traceability application. Radio frequency identification (RFID) is a wireless technology capable of automatic and unambiguous identification (without line of sight) by extracting a unique identifier from microelectronic tags attached to objects. RFID was first explored in 1940s as a method to identify allied air planes [38]. In the decades following its invention, RFID was mainly used in small-scale applications such as automatic checkouts, electronic toll collection, and anti-theft initiatives. The main reasons for RFID's limited use were the cost of RFID tags and the immaturity of the technology.

In the past decade, research initiatives by academic organizations such as the Auto-ID Center, now called the Auto-ID Labs,[1] industrial interests from companies (e.g., Wal-Mart) and government initiatives (e.g., the United States Department of Defense) have rapidly escalated new developments and interests in RFID technology. Alongside, Moore's Law has ensured that integrated circuits reduce in size, cost and power consumption. Consequently, RFID systems have become more reliable, improved in performance and more importantly, have become cheaper. These developments have resulted in an explosion in the number of RFID systems and applications deployment (e.g., tracking of tagged products in a global supply chain).

One of the important technological advances that has made this explosion possible is the so-called "Networked RFID" [46, 51]. The basic idea behind Networked RFID is to use the Internet to connect otherwise isolated RFID systems and software. Networked RFID not only eases the integration of distinct RFID systems, but more importantly, addresses the limitations of passive tags (e.g., communication, computation, and storage). The EPCglobal Network—designed by the Auto-ID Labs and developed further by the EPCglobal[2]—is a recent notable effort for Networked RFID. The EPCglobal Network is an architecture to realize a "data-on-network" system, where RFID tags contain an unambiguous ID and other data pertaining to the objects are stored and accessed over the Internet.

Significant and promising benefits from "Networked RFID" are related to enabling traceability. For example, traceability applications analyze automatically recorded identification events to discover the current location of an individual item. They can also retrieve historical information, such as previous locations, time of travel between locations, and time spent in storage. Many organizations from industry to military are planning or already exploiting RFID to enable traceability. Wal-Mart, the world's largest public corporation by revenue, in 2005, mandated its top 100 suppliers to tag their pallets and cases using RFID [7]. The U.S. Department of Defense released a policy on the use of RFID to its external suppliers and for internal operations in July of 2005 [48].

However, to reap such benefits researchers must overcome a number of key challenges. RFID traceability is not a single-layer problem. First of all, large-scale global

[1] http://www.autoidlabs.org/.

[2] http://www.epcglobalinc.org.

RFID networks have the potential to generate unprecedented amounts of data related to individual objects. An important challenge therefore centers on the efficient management and sharing of the data with traceability applications. A system architecture for data gathering, processing and sharing must also be scalable in order to deal with the data collected from networked RFID systems. For efficient processing and storage, data models must be carefully considered. To allow business users to make useful decisions and analysis, we must support different types of traceability queries, perhaps also by exploiting some high-level business logic. Finally, RFID is a pervasive technology that can unobtrusively monitor the movement of tagged goods or persons to generate sensitive data. As a result, privacy and security concerns must be addressed to allow wide-scale real world adoption.

Driven by the numerous potential application benefits and research challenges, RFID traceability networks are becoming an active research and development area [25, 46, 53, 57, 58]. Many researchers are currently engaged in developing solutions to address these challenges. In this paper, we survey the state-of-the-art solutions in realizing large-scale RFID traceability networks capable of supporting item-level (also called serial-level) traceability. To this end: (i) we derived a set of key traceability queries that an RFID enabled traceability system should be able to support, (ii) we identified central attributes that should be possessed by a system architecture in developing traceability applications, and (iii) we also identified a set of important attributes that a data model should have for efficient processing of traceability queries. These attributes are used as a benchmark to study the state of the art techniques on RFID traceability networks.

The aim of this work is to provide a better understanding of current research and challenges in the area of RFID enabled traceability networks. The scope of our work is limited to considering information system architectures and RFID data management of this increasingly active area of research. To the best of our knowledge, this is the first effort that studies the state-of-the-art techniques for RFID traceability networks.

The remainder of this paper is organized as follows. In Sect. 2, we give a brief introduction to RFID technology, overview the concept of traceability, and present several typical RFID traceability applications as powerful motivations for our work. In Sect. 3, we introduce a generic reference framework for traceability networks and examine fundamental traceability queries. In Sect. 4, we identify a set of essential system development requirements and desirable data model attributes for developing large-scale traceability networks, which will be used as a set of criteria to compare current solutions. In Sect. 5 and Sect. 6, we provide a detailed investigation of current architecture proposals and data models for RFID traceability networks respectively. We also highlight some future directions for research and development. Finally, we conclude the article in Sect. 7.

2 Background

2.1 RFID basics

RFID technology is used to create a seamless link between individual, physical objects and their digital natives. RFID allows individual objects to be uniquely and

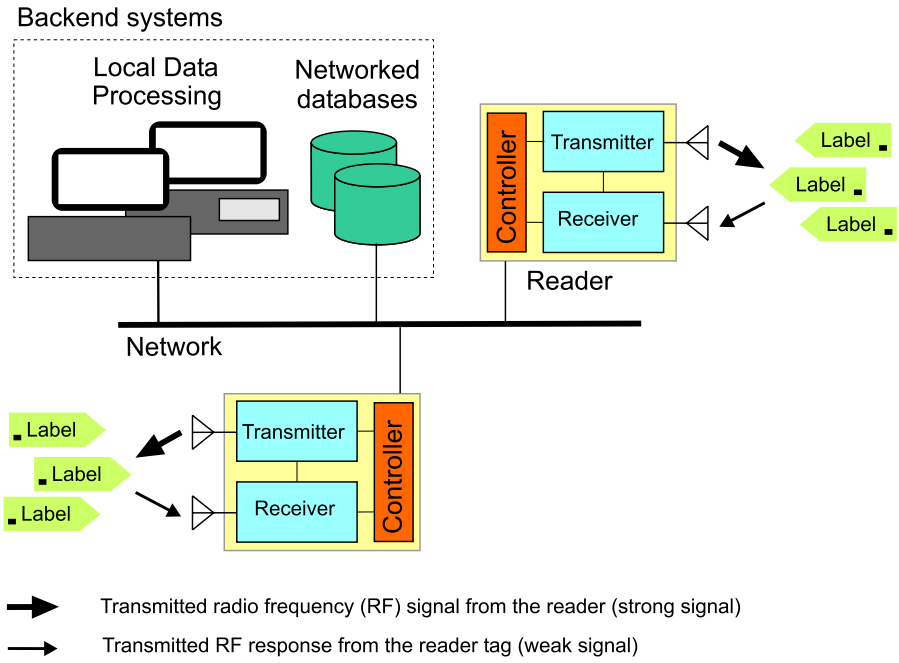


Fig. 1 Overview of an RFID system

automatically identified (Auto-ID) using wireless communications to extract identifiers from RFID tags attached to objects. In contrast to traditional identification technologies such as magnetic strips or bar codes, RFID is a contactless technology that operates without line-of-sight restrictions [22].

Regardless of the underlying technologies around which an RFID system is built (e.g., microelectronic tags, surface acoustic wave tags, tags using multiple resonances to encode data and so on), all modern RFID system infrastructures can be categorized into three primary components, namely *tags* (*labels*), *readers*, and *backend systems*. Figure 1 illustrates the interconnected components of a typical modern RFID infrastructure.

Tags Tags, also called RFID labels, are attached to objects. A tag contains an integrated circuit (IC) or a *chip* that stores the identifying information of the object to which the tag is attached as an electronic code and an *antenna* that communicates the information via radio waves. When a tag passes through an electromagnetic field generated by a reader, the tag communicates back to the reader the identifying information. Consequently, there is no line-of-sight requirement for object identification in RFID systems.

The data stored on the tag, object identifying information, may be an Electronic Product Code (EPC) [17], which is a unique item identification code. Although a variety of existing as well as hitherto undefined identification codes can be encoded as EPC, an EPC typically contains information that identifies the manufacturer, the type of item and the serial number of the item.

RFID tags can be classified based on their frequency of operation (Low Frequency, High Frequency, Ultra High Frequency or Microwave), or according to powering techniques (*passive*, *semi-passive*, and *active*) [22]. An active tag has its own transmitter and a power source to power the microchip's circuitry and broadcast signals to an RFID reader. The power source is either connected to a powered infrastructure or uses energy stored in an on-board battery. In the latter case, an active tag's lifetime is constrained by the battery. A passive tag does not have its own power source and scavenges power from the electromagnetic fields generated by readers. A passive tag also has an indefinite operational life and relies on reflecting back the electromagnetic (EM) field generated by the reader and modulating the reader's EM incident on the antenna to send information back. Semi-passive tags use their own power source to run the microchip's circuitry but scavenge power from the waves sent out by readers to broadcast their signals.

Active and semi-active tags are more expensive and typically used for high-value goods and/or large assets that need to be tracked over long distances. For example, the U.S. Department of Defense uses active tags to track many containers being shipped to bases and units overseas. On the other hand, passive tags are very inexpensive (as cheap as 20 cents) and can even be used for common materials in very large quantities. Currently, significant efforts are being undertaken to achieve *5-cent* tags by shrinking chip size, cutting antenna cost, and increasing tags consumption (e.g., RFID mandates from Wal-Mart and U.S. Department of Defense).

RFID tags appear in a wide variety of shapes (e.g., key fobs, credit cards, capsules, pads), sizes (e.g., small as a grain of rice, big as a six inches ruler), capabilities, and materials. Tags can have metal external antennas, embedded antennas, or printed antennas.

Readers The complexity, configuration and function of the readers, also called interrogators, depends on the application, which can differ quite considerably. However, in general, the reader's function is to generate an electromagnetic (EM) field to power tags (when passive tags are employed) and facilitate communication with tags.

RFID readers communicate with tags using a radio frequency interface. Either a strong energy storage field near the reader antenna, or radiating EM waves, establishes the RF interface. Communication between a reader and a tag may involve interrogating the label to obtain data, writing data to the label or beaming commands to the tag so as to affect its behavior. The readers consist of their own source of power, processing capability and an antenna. In addition, most modern RFID readers are equipped embedded systems with networking capabilities (WiFi or LAN) to allow readers to be networked with other computing hardware. Typically, readers are connected to a backend system via the networking interfaces (as outlined in Fig. 1). In this survey, we will not give a detailed review of physical principles regarding RFID hardware design. Interested readers are referred to [22, 59].

RFID readers are generally placed at fixed locations with their antennas strategically placed to detect tagged items passing through their EM field. RFID readers can read multiple co-located tags simultaneously (e.g., up to several hundred of tags per second). The reading distance ranges from a few centimeters to more than 100 meters, depending on the types of tags, the power of readers, interference from other RF devices and so on [22].

Backend systems The readers are connected to a computer network in which the data is collected and processed. This network may be limited to a single organization, or it may cross organizational boundaries to enable cooperation and sharing between business partners (e.g., manufacturers, warehouses, and retailers).

2.2 Understanding traceability

GS1,[3] a global organization dedicated to the design and implementation of global standards for supply and demand chains, proposes the definition of traceability as "the ability to trace the history, application or location of that which is under consideration" [29] (ISO 9001: 2000). Although the context of [29] is based on supply chain management, this definition is appropriately generic for other application areas.

It should be note that GS1 definition only refers to *historical* information. We argue that the ability to establish the present and predict the future state is a significant addition to traceability applications. For example, when an object leaves a location $\mathcal{L}$, the only information recorded is its last observed location (i.e., $\mathcal{L}$). There is a gap in information available about its destination or expected time of arrival. Such information would be potentially useful in making effective business decisions. Consequently, it is useful to articulate the implied meaning of traceability. We formally define the traceability as the following:

Traceability is the ability to retrieve past, present, and potentially, future information about the state (*e.g., location*) *of an object*.

Networked RFID systems have the potential to create revolutionary applications by enabling real-time and automatic traceability of individual objects. In this article, we will refer traceability systems built on networked RFID technology as *traceable RFID networks*.

2.3 RFID enabled traceability applications

The underlying identification technologies predominantly used in existing traceability applications (such as optical bar codes and human readable codes) require human operators and are labor intensive for implementation at the individual product level. Printed bar codes are also a line of sight technology, prone to failure by effects that reduce the visibility of the bar code (e.g., dust, dirt, physical tears). There are also other issues such as delays in transactions (e.g., bar codes need to be correctly aligned to be read) and identification inaccuracies due to human operator errors. Consequently, these systems have an important impact on the quality of traceability information.

Research suggests that the process of manually recording a re-usable container number and entering it into a computer before shipment is susceptible to 30% error [22]. The impact of such errors is costly. In manufacturing environments, scanning errors as a result of associating the wrong container to the processing steps can result in the whole batch of products being discarded due to quality assurance reasons. The capabilities of identification technologies and the costs involved to identify products

[3] http://gs1.org.

at instance level have prevented companies from being able to make decisions at the individual product level.[4]

However, with traceable RFID networks, object instances can be precisely and automatically monitored, and their life histories can be recorded in real-time. Traceability essentially improves the quality (accuracy and the level of detail) and timeliness of information leading to better decisions at the business and enterprise level. As a result, by exploiting traceable RFID networks' ability to precisely record and trace product movements automatically, there are many emerging advanced application scenarios, such as reducing costs of inventory errors, eliminating shrinkage, fine grain products recalls and anti-counterfeiting.

2.3.1 Eliminating inventory inaccuracies

The discrepancies between actual and recorded inventory in information systems (i.e., inventory inaccuracy) is estimated to be as high as 65% at a major retailer [14]. Despite significant investments by companies to reduce the information gap, the quality of inventory information is still poor and often leads to inefficient supply chains. A significant portion of inventory inaccuracies are related to two execution problems: *transaction errors* and *misplacement errors*.

Transaction errors occur unintentionally during various transactions such as an inventory count, goods receipt check and at the point of sale (e.g., when a variety of potato is recorded as a different kind by the sales staff).

Raman [14] reports that 16% of items at a leading retailer were missing as a result of products being misplaced at various locations in the store, storage or back room. Misplacement errors impact sales. Culprits of misplacement are not just employees but consumers who may pickup items and subsequently place them in other locations. A leading market research and advisory firm IDTechex, estimates that, annually, hospitals lose close to 15% of their assets by value and are unable to locate 15–20% of their assets resulting in additional costs of US $1,900 per nurse.[5]

Traceable RFID networks have the ability to reduce transaction errors through automatic capture of individual item level quantities and location information at various process steps. Similarly, misplacement errors can be minimized by analyzing the data gathered from tagged items movements obtained automatically from a network of readers strategically placed along the supply chain and at each business step.

2.3.2 Inventory shrinkage

Inventory shrinkage, as defined by the Efficient Consumer Response (ECR) group,[6] refers to the loss of inventory as a consequence of a combination of internal theft (e.g., employees), external theft (e.g., shoplifters), supplier fraud, and administration errors. Shrinkage results in a staggering annual loss of US $33.1 billion for

[4] http://www.scdigest.com/assets/On_Target/09-02-23-1.php.

[5] http://www.idtechex.comresearch/articles/rfid/in/healthcare/and/pharmaceutical/applications/00000518.asp.

[6] http://www.orisgroup.co.uk/blue_book.asp.

US retailers, Euro 28.9 billion for European retailers and AU $942 million for Australian retailers [6]. RFID traceability networks can improve and, in some cases, even eliminate shrinkage due to theft prevention in the supply chains. More importantly, RFID traceability networks provide us the capability to measure shrinkage accurately, which helps to pinpoint the likely causes.

2.3.3 Eliminating wastage and damage

The cost associated with food wastage is a significant problem for the food industry. For example, perishable fresh products while contributing only 30% to sales constitute 56% of the total wastage at supermarkets [37], which represents a significant opportunity for improvement. Many factors contribute to spoilage including unsuitable variations in environmental conditions during transport and handling, excessive dwell time during loading, transport, and unloading. RFID traceability networks can automatically capture movements, dwell time and condition of products, which make it possible for instant checks of freshness and identification of potential causes of spoilage.

2.3.4 Fine grain product recalls

Food and drug safety is widely regarded as a serious threat to public health globally. RFID traceability networks will ease the task of product recalls by rapidly and accurately locating specific harmful products in the event of problems such as an illness outbreak due to contaminated food. For example, countries are adopting policies and regulations requiring all cattle to be tagged to allow authorities to quickly locate the source of infected cows in the event of an outbreak of mad cow disease [48]. To achieve fine grained recalls, BT Foodnet[7] uses RFID to track products and provides a full audit trail of ingredients along the supply chain. Then only products with bad material need to be recalled, which significantly decreases the wastage.

2.3.5 Anti-counterfeiting

The International Anti-Counterfeiting Coalition[8] estimates that US $600 billion of goods, accounting for 5–7% of the world trade, are counterfeit. The impact of counterfeiting is not only limited to manufacturers and brand owners, but has serious consequences for consumers. The World Health Organization estimated that in 2003 between 5–8% of the worldwide trade in pharmaceutical is counterfeit [20]. Counterfeit medicines range from products with wrong ingredients, insufficient active ingredients or products with fake packaging to mimic a medication.

There are a variety of existing techniques for product authentication based on optical technologies such as watermarks, holograms, micro printing, and biochemical technology [8]. All these technologies have static markers that are generally applied on a uniform scale to a single class of products. However, biochemical marker tests

[7] http://www2.bt.com/static/i/media/pdf/campaigns/consumer_goods/foodnet_broch.pdf.

[8] http://www.iacc.org.

provide the ability to detect markers but they do not generally quantify the marker, thus leaving open avenues of counterfeiting by dilution. Optical technologies no longer present an adequate deterrent due to the reduction in the cost of producing imitated watermarks and holograms.

RFID enabled traceability has the potential to provide a timely and an automatic trace that can verify the existence of a valid chain of custody through a supply chain, which is commonly referred to as providing an electronic pedigree [44]. Recent legislation has even pushed industries to consider RFID technology to comply with electronic pedigree laws. For example, some states in the USA have introduced the pedigree laws [28] requiring a verifiable record of drug movement through the supply chain at any time. Furthermore, traceability data can be analyzed using machine learning algorithms to detect and report anomalies in supply chains and to alert potential problems or separate counterfeit products from genuine products using copies of genuine product identifiers [32].

3 The reference framework and traceability queries

As discussed in Sect. 2.3, RFID enabled traceability networks have many important applications. However, these applications also have different information requirements. It is therefore important to identify a generic framework and a set of fundamental queries necessary to support the development of traceability applications.

We followed a comprehensive methodology to elicit and analyze traceability application requirements in order to determine specific data management and information systems related requirements. Our approach considers (i) the responses to a survey conducted among potential end-users and vendors in Australia[9] (the primary goal of the questionnaire was to extract requirements for traceability applications), (ii) the evaluations of the outcomes of the EU funded BRIDGE project,[10] which aims at developing a traceability platform based on identified industrial requirements from enterprises in Europe, and (iii) our analysis of existing literature on traceable RFID networks [11, 19, 20, 27, 29, 32, 36, 39, 46, 49, 56].

3.1 The reference framework

To ease our discussion and better understand a traceability network and its elements, we propose a generic reference framework (Fig. 2b) that is agnostic to various traceability applications by modeling elements of a traceable RFID network. Figure 2a shows an example of a small supply chain network. It is modeled in Fig. 2b using the reference framework which consists of the following components:

Node. Nodes represent observation points in a traceable RFID Network. A node can be a geographic location of an organization or an internal location within an organization. Physically, each node may represent an RFID reader antenna installation to

[9] http://cs.adelaide.edu.au/peertrack/collaboration/survey/.

[10] http://www.bridge-project.eu/.

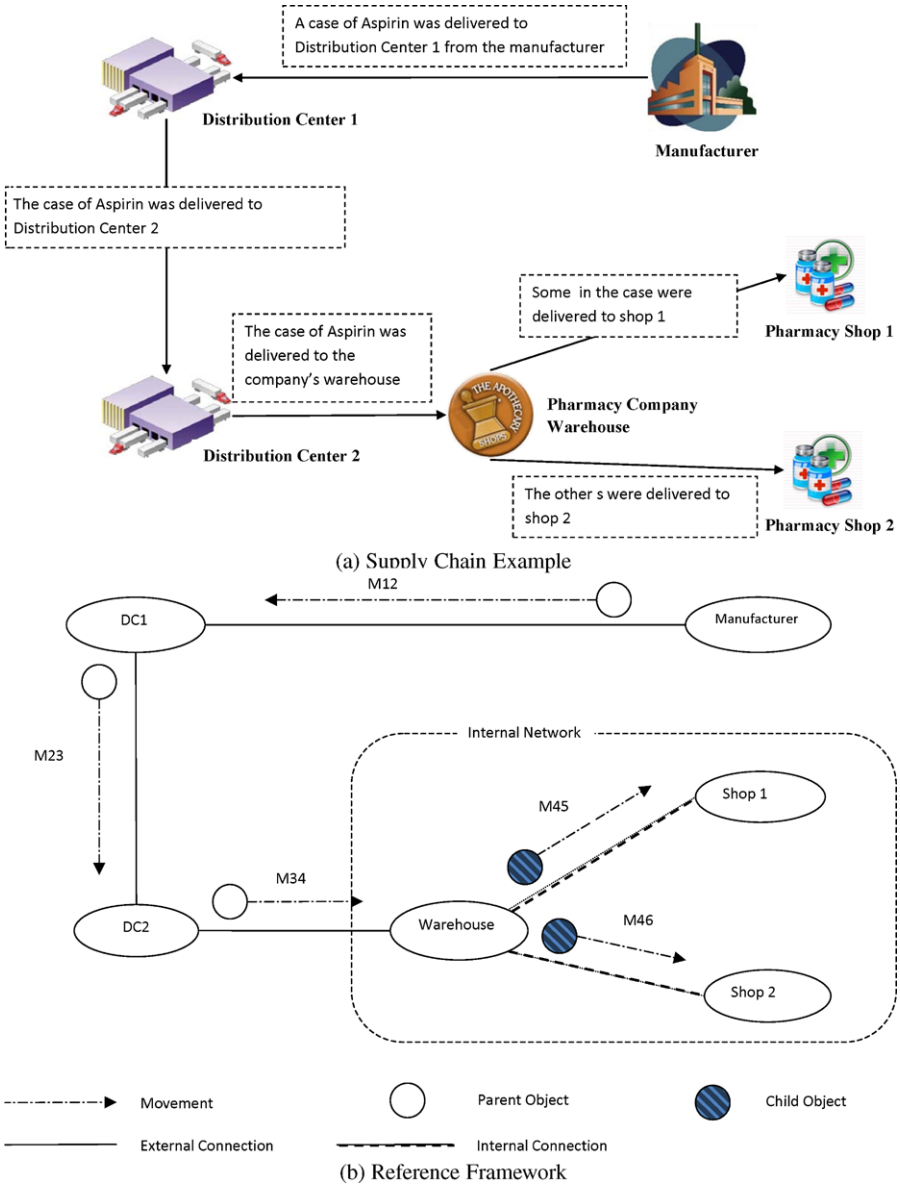


Fig. 2 Example of reference framework

collect and forward RFID data associated with *objects* passing through its detection area. But not all the physical locations with RFID reader(s) may formulate as a node in the traceability network. The number of nodes will vary with user requirements and the location granularity level. For example, in a supply chain, the internal flow of goods inside a distribution center may not be of any interest to other trading partners while it may be critical for the distribution center to manage its inventory. As a

result, for the partners, observation points in the distribution center do not count as *nodes*, while for the distribution center, they do.

Object. An object represents a tagged item with a globally unique identifier.

Connection. A connection is a link between nodes. It is established statically (e.g., by the partnership of organizations). However, it is *quasi-static* because these partnerships or supply paths may change over time. Each connection may be characterized by several properties or meta data (e.g., distance to neighboring nodes, possible methods and cost of travel).

Network. A network is a set of composite *connections* that is quasi-static. It represents the direct or indirect relationship between *nodes*. According to data sharing policies, networks are categorized into two types, *Open-Loop* networks and *Closed-Loop* networks. Within a Closed-Loop network, data is shared by nodes that belong to the same organization. On the other hand, nodes in an Open-Loop network normally belong to different organizations.

The following lists several concepts that encapsulate the dynamic relationship of objects in a traceability network.

Movement. This captures the movement of an object from a source node ($\mathcal{N}_s$) to a destination node ($\mathcal{N}_d$). A movement can be represented by the triplet $\{\mathcal{N}_s, \mathcal{N}_d, \mathcal{T}\}$, while $\mathcal{T}$ represents the time taken for the movement.

Dwell. The time an object remains at a node.

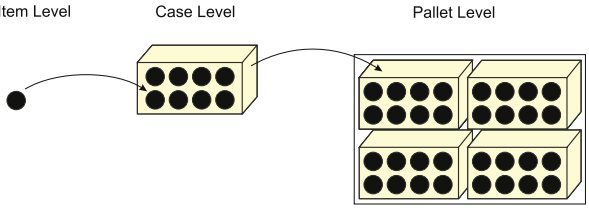
Path. A set of ordered movements establishes a path (e.g., $\{\mathcal{M}_{12}, \mathcal{M}_{23}, \mathcal{M}_{34}\}$ in Fig. 2) through the network. Paths are records about the history of an object in both spatial and temporal dimensions.

Containment. Objects may be organized hierarchically. A parent object can contain one or more child objects. This relationship is known as *containment* in our discussion. The containment of objects may be changed between movements. Child objects may be separated from a container (i.e., the parent object) at some point or some objects may join a container. These changes of the containment relationship must be carefully managed in order to be able to respond to traceability queries. Containment is modeled by the hierarchical structure as shown in Fig. 3.

3.2 Traceability queries

It is difficult to determine exact query requirements because they are largely application dependent. A common application oriented classification of traceability queries proposed in [11] includes: (i) *pedigree queries* that reconstruct the complete historical path of an object through a supply chain, (ii) *product recall queries* that detect the

Fig. 3 Containment hierarchy



current location of objects, and (iii) *bill-of-material queries* that return information about all the objects with a containment relationship of a specific object. However, this is not an adequate generalization for supporting traceability applications.

In this survey, we formulate a set of fundamental queries that are useful for most traceability applications, which can be used as building blocks to construct more complex queries. It is evident that the information critical to the success of a traceability application is the determination of an item status (identity, precise location, physical status such as perished/damaged/expired etc.) and history of its path throughout the supply chain. The key objective of traceable RFID networks is to enable the discovery of past (trace), present (track) and possibly future information (prediction) of objects. Consequently, tracking, tracing and prediction are the three fundamental types of traceability queries that can be generalized as being adequate for building traceability applications. Tracking refers to a query to find the current state (such as its current location) of an object. Tracing refers to finding the historical states of an object and prediction queries provide a probabilistic view of possible future states of an object (e.g., the most probable node to be visited next). We summarize and categorize these queries as the following:

Track queries A track query supports the retrieval of the current state, such as the location, of an object. The following is a typical example of a track query:

Q1: Where is object $\mathcal{O}_x$ *now*?

Trace queries These queries are designed to discover a part of or the whole life history of an object such as movement information. Some typical examples are:

Q2: What nodes did object $\mathcal{O}_x$ travel through before it reached node $\mathcal{N}_y$?
Q3: What is the travel path for object $\mathcal{O}_x$?
Q4: What is the travel path for object $\mathcal{O}_x$ before it reached node $\mathcal{N}_y$?
Q5: What are the nodes visited by object $\mathcal{O}_x$ after node $\mathcal{N}_y$?

RFID event data is characterized by both spatial and temporal information [56]. As such trace queries can be spatially or temporally constrained to discover the location of a given object at a specified time or determining places where the object has been for a particular time period. Some typical examples are given below.

Queries with *spatial constraints*:

Q6: Where was object $\mathcal{O}_x$ on *12th, Dec. 2010*?
Q7: Where has object $\mathcal{O}_x$ been between *12th, Dec. 2010* and *20th, Dec. 2010*?

Queries with *temporal constraints*:

Q8: When was object $\mathcal{O}_x$ seen at node $\mathcal{N}_y$?
Q9: How long did object $\mathcal{O}_x$ dwell at node $\mathcal{N}_y$?
Q10: How long did object $\mathcal{O}_x$ take to move from node $\mathcal{N}_x$ to $\mathcal{N}_y$?

There are a specific set of trace queries aimed at extracting containment relationships between objects such as finding the objects that traveled in a pallet. These are called *containment trace* queries. Some typical examples are:

Q11: What objects were contained in object $\mathcal{O}_x$ on *12th, Dec. 2010*?
Q12: What was the container for object $\mathcal{O}_x$ when it was at node $\mathcal{N}_x$?
Q13: Where was object $\mathcal{O}_x$ packed into object $\mathcal{O}_y$?
Q14: When was object $\mathcal{O}_x$ unpacked from object $\mathcal{O}_y$?

Finally, a set of beneficial trace queries to support business processes and strategic decisions are desirable to data analysis applications (e.g., ERP applications designed to manage the re-ordering or production of goods). As such, in a supply chain, it is useful to know the average time spent by various products in storage at a particular distribution center or on a shelf at a supermarket. These queries are classified as *statistical trace* queries. Some typical examples are:

Q15: How many objects have been sent from node $\mathcal{N}_x$ to node $\mathcal{N}_y$ last year?
Q16: Which node sent node $\mathcal{N}_x$ the maximum number of objects last year?
Q17: Which node received the minimum number of objects from node $\mathcal{N}_x$ in 2010?
Q18: What is the average dwell time of object at node $\mathcal{N}_x$?
Q19: What is the total number of objects seen at node $\mathcal{N}_x$ in 2010?

Prediction queries The objective of a prediction query is to estimate the future state of an object. Some typical examples are:

Q20: What is the expected arrival time for object $\mathcal{O}_x$ at node $\mathcal{N}_x$?
Q21: What is the probability that object $\mathcal{O}_x$ will arrive at node $\mathcal{N}_x$ in the next hour?
Q22: What is the expected location of object $\mathcal{O}_x$ after node $\mathcal{N}_x$?
Q23: What is the expected location of object $\mathcal{O}_x$ after five *movements* from node $\mathcal{N}_x$?

4 Traceable RFID network requirements

There are different approaches, as discussed later, to realize RFID enabled traceability applications. These approaches differ in many aspects such as hardware used, system architectures, and data models. Although traceability is a multi-layer problem, successful traceability systems must be able to satisfy a number of key system development requirements such as scalability and timely responses. Another significant aspect to realizing traceability applications is the development of novel data models to support efficient processing, storage and retrieval of information from RFID event data. In this section, we identify a set of dimensions for evaluating existing traceability approaches. We consider the dimensions based on system development requirements and data model requirements.

4.1 System development requirements

We consider the following key system development requirements: (i) *support for unique identifier*, (ii) *uncertainty*, (iii) *support for prediction query*, (iv) *scalability*, (v) *heterogeneity*, (vi) *timely responses*, and (vii) *security and privacy*.

4.1.1 Unique identifier

Given the distributed nature of data collection and storage, there must be a mechanism for associating products with their relevant life-cycle data in networked information systems as well as on products themselves. This aspect is fundamental to networked RFID systems. The unique identifier (UID) forms the link between an object and its associated information collected and possibly distributed at various organizations and locations [45]. The UID can then be used to discover and access information associated with the UID from distributed information resources, similarly to the manner in which web addresses or Uniform Resource Locators (URLs) are used to access information from the Internet.

Supporting traceability applications such as targeted product recall and anti-counterfeiting requires that each architecture supports a unique identifier. The scope of the identifier may be defined by the application. However, for managing global traceability applications with a worldwide focus (such as supply chains distributed across countries), a fundamental requirement is the support for a globally unique identifier.

4.1.2 Uncertainty

The responses to traceability queries (see Sect. 3.2) may not be deterministic since the underlying RFID network is limited by the number of discrete observation points (nodes), hardware performance, and data sharing issues. Consequently, a significant challenge is to overcome uncertainty. There are different causes for uncertainty:

False positives. Data generated by readers is limited in accuracy. RFID readers may report a tag identifier that is not stored on a tag within the reader's EM field (ghost reads) due to various reasons outlined in a research report from Auto-ID Labs.[11] Essentially, the reader receives incorrect data which is interpreted by the reader as being valid. Device malfunctions may also produce incorrect data to be forwarded by readers to backend systems. False positives result in erroneous data that is difficult for information systems to handle.

Missing events. A reader may miss identifying an object or a temporal malfunction of a device may cause a systematic error in events generated at a node. A missed tag read results in no data since data, such as the identifier stored on the tag is not captured by the reader.

Nodal limitations. At a given time, an object may be in transit between two nodes or it may have arrived at a node but yet to be identified. If we consider *Q1*, the answer might be the source node but in fact the object already left the source node either in transit or at the destination node. This detachment of the digital observation from physical reality applies to all other queries.

Containment visibility. A special case of uncertainty is introduced through data sharing issues in an open-loop network when containment relationships are involved. For example, if a node at a partner organization does not release containment relationships involving child objects joining or leaving parent objects, it may not be

[11] http://www.autoidlabs.org/single-view/dir/article/6/77/page.html.

possible to obtain accurate information regarding the complete life history of a specific object.

Certain causes such as false positives can be eliminated with hardware improvements. For example, the false positives problem has been solved through second generation RFID readers compliant with EPCglobal's Class 1 Generation 2 air interface protocol [18]. However, uncertainty cannot be completely eliminated (e.g., missing events, nodal limitations) via hardware improvements alone and must be dealt within the software layer. Therefore factors that affect the accuracy and completeness of data must be addressed in RFID traceability systems to ensure the validity of responses to queries.

4.1.3 Prediction query support

The ability to predict future states (such as the estimated arrival time to the next node) of individual objects is an important aspect of dynamic planning and control as well as scheduling of shared resources. For example, with the ability to predict the network status, such as possibility of congestions at a node or a connection, managers can resolve potential supply chain disruptions beforehand. In such an environment, key business decisions are driven by information related to an estimated arrival time of an object at given destinations or the *probability* of arriving within a certain time frame. Analysis of past performance metrics and data mining techniques can be used to detect patterns and estimate future movements based on data collected by traceable RFID networks.

4.1.4 Scalability

In large-scale RFID applications (e.g., global supply chains), there will be thousands of readers distributed across and within organizations that generate large volumes of data automatically and rapidly. Data volumes can be enormous (e.g., Wal-Mart generates about 7 tera-bytes of data every day if goods are tagged at the item level [52]). A scalable architecture framework is required to ensure adequate performance of traceability networks as the number of nodes and volume of data increases. A scalable architecture must address the following issues:

Data volumes. Given the large quantities of potential object instance level data, an appropriate solution that does not involve the permanent storage of individual raw data must be found.

Integration. It should be possible to integrate increasing number of nodes into the traceable RFID network without degrading query performance such as timeliness of responses. This is significant since a linear increment of the number of nodes will also linearly increase the number of nodes that must be searched for object related data in a blind search.

4.1.5 Heterogeneity

A traceable RFID network is established by connecting different nodes, which may belong to different organizations, use different hardware and software systems, store

the collected data in different formats. In addition, with the rapid development in RFID technologies, new devices may be introduced. Consequently, traceability systems should be agnostic to such heterogeneity and, ideally, be compliant with global standards for interoperability across of organizations and geographies.

4.1.6 Timely responses

Traceable RFID networks are built on the premise that changes in the physical world are reflected by timely changes in information systems. Real-time traceability information is critical for managing distribution operations, rapid product recalls and service/maintenance operations that need to be constantly re-evaluated based on traceability information of tools and technicians. Therefore, an expectation of a traceable RFID network is that the system should be responsive, with the ability to provide timely information.

4.1.7 Security and privacy

RFID is a pervasive technology capable of mass serialization and unobtrusive scanning from a distance. So no discussion is ever complete without addressing various security and privacy related issues. Traceable RFID networks are susceptible to issues arising from vulnerabilities in RFID technology [34, 44] as well as associated information systems. For example, competitors of an organization (such as a rival supermarket) may scan another organization's inventory labeled with RFID tags or eavesdrop on the organization's own valid operations to obtain valuable information, such as sales data, to ascertain the performance of its competitors (an act commonly referred to as corporate espionage). The fact that a third party can eavesdrop on a conversation between a tag and reader from a distance is a fundamental vulnerability.

There are numerous publications [5, 16, 21, 35, 42, 44] that address vulnerabilities of RFID systems through improved security features such as the kill functionality for Class 1 Generation 2 tags [17] and lightweight security mechanisms suitable for RFID devices. Furthermore, there is a mature and standardized set of cryptographic tools (e.g., public key security mechanisms such as RSA and Elliptic Curve Cryptography, private key mechanisms such as the Advanced Encryption Standard) available for securing computer networks and networked resources. Therefore, we will only consider the traceability system's ability to manage RFID data without violating privacy or compromising security of partner organizations participating in a traceable RFID Network.

In RFID networks, objects that move among nodes belonging to different organizations leave partial information related to their life-cycle at each node. To achieve the full potential of traceability applications, partner nodes in a traceable RFID network must share collected data with other trading parties. However, for competitive reasons such as surreptitious collection of product related sales data from competitors or security reasons, most organizations are reluctant to share their data. The competing needs between traceability and data privacy are unavoidable issues in networked RFID systems across organizations. Nevertheless they must be managed successfully to ensure that solutions developed are implemented in practice.

4.2 Data model requirements

Most modern RFID readers collect data from tag reading events as a data triplet {id, time, node}. The temporal and spatial information is implicit. For example, to discover the dwell time of object $\mathcal{O}_x$, an infant formula, at node $\mathcal{N}_y$, a storage shelf at a distribution center (such as Q9), we have to sort all reads for $\mathcal{O}_x$ at $\mathcal{N}_y$ in time and the dwell time is obtained by the time difference between the first and the last reads. The level of pre-processing required to respond to such a query makes the processing of this query inefficient. In particular, for large-scale systems, it will result in severe performance issues. Most traceability queries are in fact much more complicated.

To make query processing more efficient, a high level data model that considers the characteristics of queries is required. We propose the following requirements for a suitable data model based on the traceability queries discussed in Sect. 3.2:

Temporal abstraction. RFID data is generated dynamically and associated with timestamps. It is highly desirable for the model to abstract temporal information from the underlying data, such as dwell time, time taken for a movement, and arrival and departure time. Such high-level temporal attributes will vastly improve the processing of trace queries such as Q8 and Q10 by reducing the number of basic queries that needs to be executed as well as by eliminating the time required to process the related responses to derive a final response.

Spatial abstraction. A node is associated with a location. Many traceability queries are related to discovering movements of objects between nodes (e.g., Q2–Q7). Similar to temporal abstractions, it is desirable for the data model to be able to provide a high level representation that captures object movement, path and other spatial information to efficiently process trace queries.

Containment relationships. The data model should be able to encapsulate changes in containment relationships as objects move across nodes. In other words, the model should be able to preserve the dynamic relationships between parent and child objects. For example, individual items are packed in cases and pallets, which are then unpacked or repacked in new pallets. With containment relationship captured in the data model, queries Q11–14 can be effectively processed. In addition, the storage cost can be significantly reduced by grouping the records.

Statistical constructs. To efficiently process statistical trace queries (see Q15–Q19), it is highly desirable for the data model to be able to encapsulate low-level statistical information such as sums and averages based on RFID event data collected by the system. Although this is not an essential attribute, processing a statistical query such as Q16 over a traceability system with a complex supply network structure (multiple pathways into and out of a node) would not be possible using a set of low level trace queries we have discussed in Sect. 3.2.

Uncertainty. We have discussed the need for addressing uncertainty introduced as a result of the imperfections in the physical layer in Sect. 4.1.2. In addition to information systems' support, appropriate data models are required to capture and model the uncertainty in the status of objects derived from observed events and the actual status of objects in the physical world.

A good data model lays the foundation for efficient traceability query processing. However, for the data model to be easily integrated into a traceable RFID network,

there are also other relevant considerations that are not directly linked to supporting traceability queries. The most significant one is *storage efficiency*. RFID data can be voluminous in large-scale applications. Storage requirements become a crucial issue for large quantities of RFID data which need to be accessed rapidly to support real-time performance of traceability applications. Good data compression methods can effectively and efficiently decrease the processing time and storage footprint.

5 State-of-the-art architectures for traceable RFID networks

In the past decade, the rapid deployment of RFID technology is making the collection, processing, integration and sharing of RFID data an active area of research and development [46, 57, 58]. In this section, we present an overview of current efforts being developed to achieve traceable RFID networks. We study these approaches and compare them using the system development requirements outlined in Sect. 4.1 and consider their support of track, trace and prediction queries identified in Sect. 3.2. Based on the analysis, we also point out some challenges and open issues that need to be addressed.

5.1 EPCglobal architecture framework

EPCglobal [17] is an organization focusing on developing standards to support RFID in information rich trading networks. EPCglobal Architecture Framework (EAF), illustrated in Fig. 4, is widely regarded as one of the most well-known RFID network architectures in industry [18]. EAF is a collection of standards for hardware, software and data interfaces (i.e., "EPCglobal standards"), together with several core services (i.e., "EPC Network Services") as shown in Fig. 4. This framework is a layered architecture that separates functionalities into three isolated modules, namely *identity*, *capture*, and *exchange*.

Identity The identity layer standardizes data representation in RFID tags (i.e., "EPC Tag Data Specification" and "Tag Data Translation" in Fig. 4). An important standard in this layer is the Electronic Product Code (EPC) defined in the Tag Data Specification.

EPC is designed to be a scalable license-plate identification number that enables linking between an individual product and its associated information resources or backend information services. EPC product identifiers can be formatted as URNs (Universal Resource Names) for use in the EPC network as described in the "EPC Tag Data Standard" (a ratified open standard).

EPC achieves uniqueness by delegating the responsibility for blocks of its number space (EPC Manager numbers) to particular companies, while guaranteeing uniqueness globally by central management of the allocation of EPC Manager numbers, to ensure that only one company would be assigned an EPC Manager number. Furthermore, in an effort towards coherence, the existing family of GS1 coding schemes (e.g. serialized GTIN—Global Trade Identification Number, SSCC—Serial Shipping Container Code, or GRAI—Global Returnable Asset Identifier) can be expressed within the EPC format.

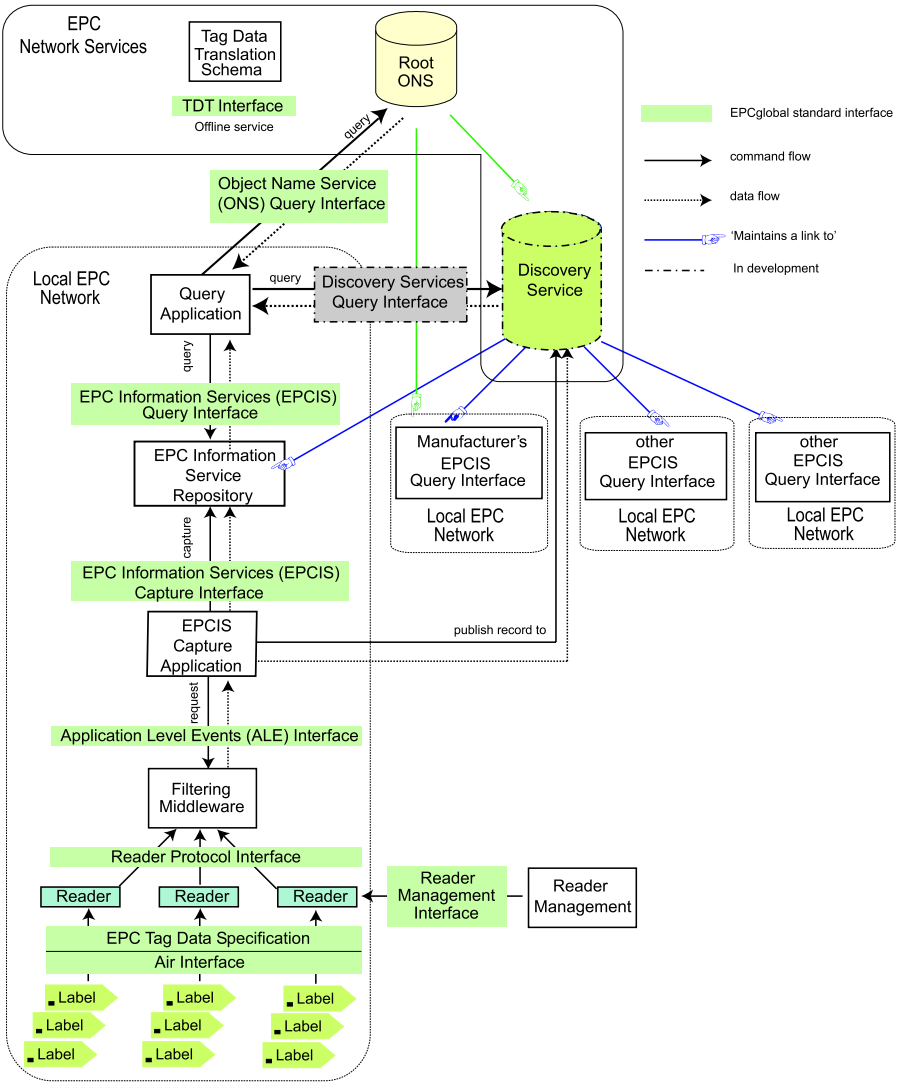


Fig. 4 EPCglobal architecture

The air interface standards define the specifications for data and commands to be transferred between tags and readers (e.g., the Class 1 Generation 2 UHF Air Interface Protocol Standard or "Gen 2"[12]).

Capture The capture layer consists of standards for reader management, reader protocols and most importantly, the Application Level Event (ALE) interface. ALE is a common interface for accessing processed RFID data and controlling the collection

[12] http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2.

of raw RFID event data sent from RFID readers. The ALE specification describes the behavior of aggregating and filtering of RFID data within a period of time. This period of time is called *Event Cycle* (EC) which can be defined in different ways:

Time-based. The boundary is simply a certain number of milliseconds to wait before stopping the processing.
Stability-based. The processing continues until stability in the set of tags is detected.
Trigger-based. This boundary allows an external controller to start and stop the process.

At the end of an event cycle, the data collected is processed and transformed into a report with filtered event data containing *what*, *when* and *where* information. ALE also defines these report specifications. An example of an ALE implementation, referred to as the "iMotion Edgware Platform", is documented by Globeranger [24] and a comparison of ALE implementations with other middleware designs can be found in [4].

Exchange The data exchange layer is designed as a service-oriented architecture [43, 45]. In this layer, three kinds of services are defined to enable data provisioning and discovery:

EPC Information Service (EPCIS). EPCIS is the first step to enable data sharing and object tracing between partners. It defines a set of roles and interfaces for data capture and query. It also defines a high level data model to classify data as either Master Data or Event Data. Unfortunately, it does not specify how these interfaces should be implemented. The cooperation and data sharing methods between partners are defined via two interfaces, namely *EPCIS Query Control Interface* and *EPCIS Query Callback Interface*. The former defines operations that can be used by partners to obtain processed RFID data, while the latter is used to obtain the data immediately after it is captured. Both interfaces define access control policies to allow only authorized trading partners to access data.
Object Name Service (ONS). The ONS functions like a "reverse phone directory" since the ONS uses a number (EPC) to retrieve the location of EPC data from its databases. The ONS is based on existing DNS systems and thus queries to, and responses from, ONS adhere to those specified in the DNS standards (RFC 1034: Domain names, Concepts and facilities). In fact, ONS uses a particular type of DNS record, called Naming Authority Pointer (NAPTR) records, (defined in IETF RFC 2915), to provide for future flexibility, since NAPTR records support the use of regular expression pattern matching; although this is not currently used in ONS, it means that the ONS results can in future be interpreted as a pattern match, resulting in a URL address which contains the serial number as part of the URL, without needing to add an ONS record for each serial number of a particular product type. The main purpose of this service is to provide a pointer to authoritative product information resources for a given EPC of an object. Consequently, the ONS only resolves EPCs to the nodes that originally assigned the EPC codes to the product (the manufacturer of a product). NAPTR records from the ONS can be service descriptions, for example, in the form of Web Services Description Language (WSDL) files, a Universal Resource Locator (URL) to an EPCIS server or an Hyper Text Markup Language

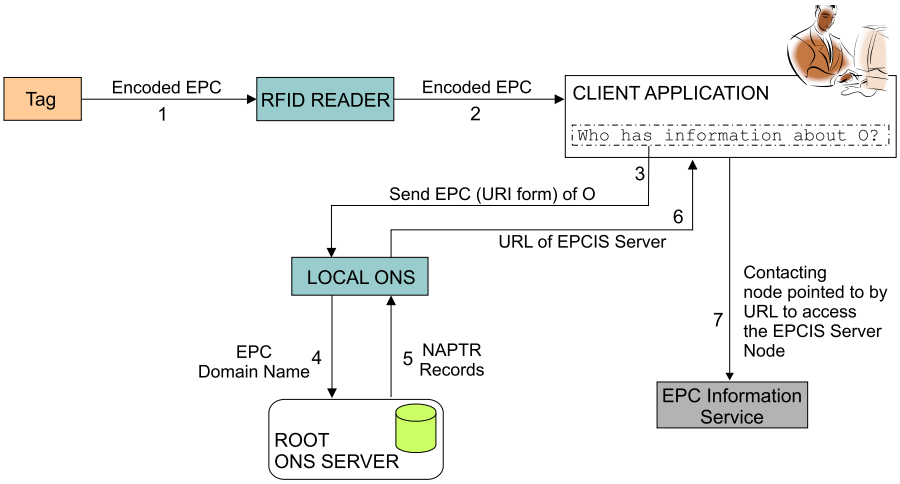


Fig. 5 EPCglobal network example

(HTML) web page. Access to ONS records is also controlled by strict access control mechanisms.

Discovery Services (DS). This module is currently not implemented and EPCglobal is, at the time of writing this article, gathering user requirements as a first step to developing Discovery Services.[13] This module is expected to discover information for a specific object, which may be distributed across many nodes among a chain of trading partners.

For example, as illustrated in Fig. 5, the client wishes to initiate a query to obtain information regarding an object $\mathcal{O}$: (1) a reader interrogates the tag on object $\mathcal{O}$ and obtains the EPC in binary form, (2) the EPC obtained (as a binary number) is passed to the client application, which is then converted into URI form to provide a means by which the client application is able to manipulate the EPC codes independent of any tag level encoding scheme. All URIs are represented as Uniform Reference Names (URNs) using the standard format defined in RFC2141 using the URN Namespace `epc`. URI is converted into domain name form so that a query in the form of a DNS query for a NAPTR record for that domain can be issued. (3)–(4) then a host lookup for NAPTR records is performed. In the event that the local ONS can not respond, the request is sent to the global ONS server infrastructure, (5)–(6) the ONS server infrastructure will return a set of possible URLs that point to one or more services, (7) the correct URL is picked and extracted from NAPTR records by the application depending on the application type and need. Finally, the client application contacts the desired service.

EPCglobal Architecture Framework aims to develop a complete, scalable and extensible specification for global trading networks. Provided that in the near future the role of discovery services and interfaces are standardized, EAF will meet many of the requirements, albeit to different degrees, that we have discussed in Sect. 4.1:

[13] http://www.gs1.org/gsmp/kc/epcglobal/discovery.

Unique Identifier. The architecture is built around the Electronic Product Code standard, which supports unique identification.

Uncertainty. This aspect is not addressed explicitly in the architecture or standards.

Prediction Query Support. Prediction queries are also not supported.

Scalability. The lower layers define the modules that work in a single node. In this way, data processing is distributed across individual nodes. The upper layers are defined as interfaces with increasing levels of abstraction. EPCIS, which is used to expose an organization's data to other trading partners, can represent a small local EPC network. In this way, a large-scale RFID network can be built by integrating many such local area RFID networks through the ONS and DS. However, the ONS is designed similarly to Domain Name Service (DNS). The problem with this architecture is two-fold: (i) the single node problem exists, and (ii) the root ONS node is susceptible to overload.

Heterogeneity. EAF is adaptive for heterogeneous organizations because of the ratified standards that are independent of a particular technology and the use of existing standards in its definitions (such as XML). By providing common interfaces, EAF successfully decouples the implementation details from the underlying hardware and software.

Timely Response. EAF is capable of supporting real-time data requirements. ALE and EPCIS support a "publish and subscribe" mechanism, whereby a client application may subscribe to a particular stream of filtered data or a particular EPCIS query which has been already defined, with the assurance that any newly received data which matches the criteria of the filter or query will be automatically sent to the client application.

Security and Privacy. Security features are either built into the standards, or use of an industry best security practice that is in accordance with EAF is recommended. For example, new ONS entries are added using a manual process where requests are submitted electronically via a web interface. These submissions are protected by ACL (access control list) and passwords. Furthermore, the EPCglobal Architecture Framework allows the use of a variety of authentication technologies across its defined interfaces. It is expected, however, that the X.509 authentication framework will be widely employed.

Data Model. An abstract data model is defined in the framework. It is more generic which makes it flexible. However, it is not completely extensible by mechanisms other than revisions to the framework [18]. For example, supporting sensor data requires extensive revisions to existing standards and cannot be simply supported by extending the data models [41]. Furthermore, temporal abstractions, spatial abstractions, statistical constructs as well as storage efficiency are issues not addressed by the EAF.

Traceability Queries. The interface standards do not provide specific traceability queries. However, access methods available from interfaces such as the EPCIS and the ONS interface gives complete freedom for applications to define and implement the application specific queries. Thus assigning more responsibilities to application developers. Taking the scenario in Fig. 5 for example, in order to get a product related information from and authoritative source the query to ONS alone requires several intermediate steps.

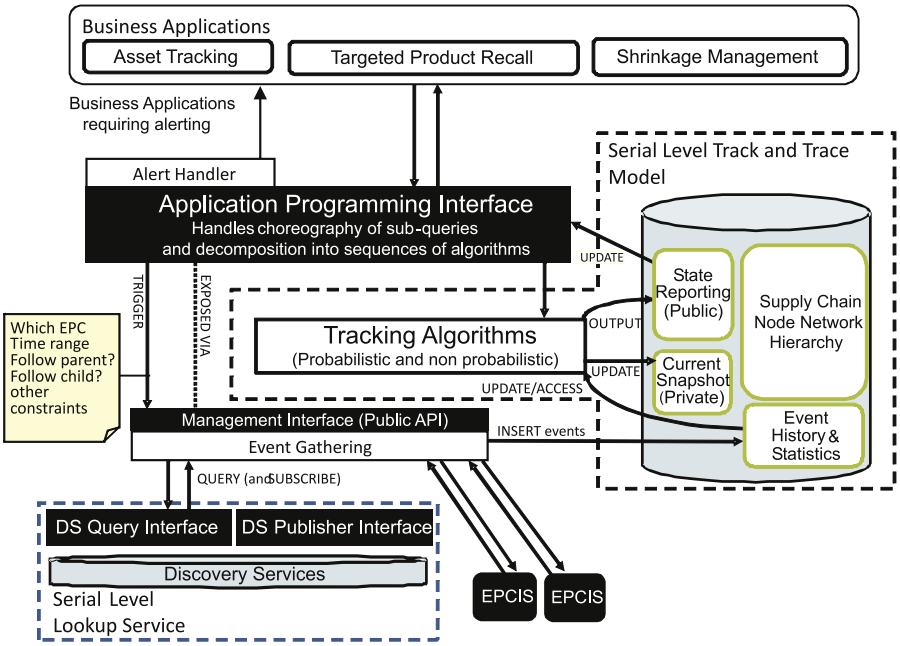


Fig. 6 The architecture of BRIDGE project

5.2 BRIDGE

Building Radio frequency IDentification for the Global Environment (BRIDGE)[14] is an European Union funded project to develop networked RFID systems. Although BRIDGE utilizes the EPCglobal standards, it explores many related fields including hardware, software and security with extensions. Work carried out within the BRIDGE project includes the implementation of the EAF (see Sect. 5.1), development of prototype Discovery Services, definition of essential interfaces such as DS publish interface, and development of algorithms and tools for building traceability applications. The BRIDGE project has explicitly taken track and trace into consideration and designed specific services for traceability queries while taking into account uncertainties. A number of successful industrial trials have been achieved in the project.[15]

BRIDGE implements traceability functionality based on the EAF using a combination of a *Serial Level Lookup Service* and a *Serial Level Track and Trace Model* (see Fig. 6). To support the Serial Level Lookup Service, BRIDGE has leveraged and extended the EPCglobal standards by developing a *DS Query and Publisher* interface along with the development of Discovery Services. The BRIDGE project has

[14] http://bridge-project.eu.

[15] http://bridge-project.eu/index.php/bridge-public-deliverables/en/.

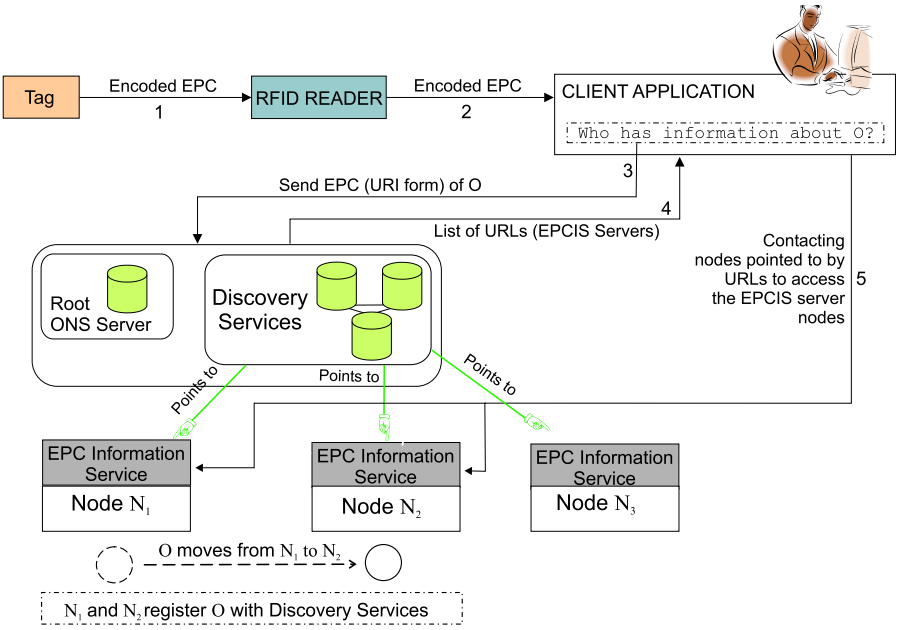


Fig. 7 BRIDGE example

addressed an existing gap in the EAF by developing the key services necessary to enable traceability applications.[16]

An important part of the *Serial Level Track and Trace Model* is a *Supply Chain Node Network Hierarchy Model* that encapsulates a supply chain model for capturing physical flow of objects in terms of nodes and connections. The track and trace model is used to model the actual state of an object and the observed state reported by the RFID devices based on a hidden Markov model (HMM), which distinguishes between the actual and the latest observed state. The HMM model describes the uncertainty under which the observed state reflects the actual successive states of the object. The *Tracking Algorithms* which consist of probabilistic and non-probabilistic algorithms provide procedures for track, trace, and prediction queries. Probabilistic algorithms are particularly designed to address uncertainty of reported observations (e.g., missing reads).

Finally, the *Event Gathering* module gathers information distributed across network nodes and the *Application Programming Interface* provides a high level interface to the serial level track and trace functionality and the tracking algorithms to support various traceability applications.

Figure 7 illustrates the critical role of discovery services within the Serial Level Lookup Service to enable traceability queries. When new information related to an object $\mathcal{O}$ is captured (e.g., moving from node $\mathcal{N}_1$ to $\mathcal{N}_2$), and stored by the EPCIS

[16]Researchers from IBM Almaden Research Center have also developed prototype DS [47]. However, the BRIDGE project in particular has demonstrated the use of DS along with the EPCglobal architecture in various industrial projects.

repository (e.g., $\mathcal{N}_1$ and $\mathcal{N}_2$), a record indicating the availability of data related to $\mathcal{O}$ is published via a DS-Publish Interface. When an Event Gathering module (see Fig. 6) requires information regarding the object $\mathcal{O}$, a query is sent to the DS, which replies all the nodes reported to contain information regarding the object (in this case the EPCIS of Node $\mathcal{N}_1$ and node $\mathcal{N}_2$). It should be noted that BRIDGE has not yet defined how the address of the initial Discovery Service is found. BRIDGE simply expects that either: (i) it is known a priori, or (ii) it is found using the services provided by the ONS.

Architecture extensions developed within BRIDGE certainly overcome some limitations of the EPC Network (e.g., uncertainty, discovery services and supporting interfaces) and meets all the requirements already satisfied by the EPCglobal Architecture Framework (e.g., explicit support for a unique identifier, heterogeneity, security and privacy). Security is even further strengthened in BRIDGE through strict access control policies governing published records as well as parties authorized to publish to DS. This is beneficial to prevent corporate espionage and surreptitious collection of business sensitive information. Limitations of BRIDGE are as the following:

Uncertainty. The track and trace model based on HMM overcomes uncertainties. This is handled in BRIDGE based on a *static* business model (i.e., the Supply Chain Node Network Hierarchy model) and a learning phase used to establish the network parameters such as transition probabilities for objects that move across a network of nodes. Consequently the model is hard to adapt to dynamic changes in the physical world (e.g., additions or removals of supply network nodes).

Prediction Query Support. Prediction query support is based on supply chain network parameter collected during a learning phase in conjunction with the static Supply Chain Node Network Hierarchy model. Consequently the usability of prediction query responses is in unclear and has not been investigated carefully within the BRIDGE project.

Scalability. The Discovery Services model has been selected specifically to allow parties on a traceable RFID network to exercise fine grain control over data related objects. However, the need for access control policies for individual object instances and the volume of potential records for the DS to provide an adequate level of support for serial level (item level) traceability queries raises concerns regarding the scalability of the approach. In particular, the access control policies are complex to manage given multiple trading parties and billions of units of items moving through a network.

Data Model. BRIDGE defines additional data model, the Supply Chain Node Network Hierarchy model, on top of EPC Network. By encapsulating prior knowledge about the supply network, this model does not maintain dependencies of nodes. This might be a significant problem since all supply chain partners must maintain an individual supply network model and communicate physical changes to all other parties in an off-line manner, or the Serial Level Track and Trace Model components must be implemented in a set of resources shared by all parties. Such an unprecedented degree of collaboration may not be desired by businesses in practice.

Traceability Queries. The track and trace model and the supply chain network structure model are used to answer track, trace and prediction queries. More specifically, BRIDGE supports track queries, trace queries with spatial and temporal constraints,

containment trace queries and prediction queries. Other types of queries (e.g, statistical trace queries) are not supported explicitly. Execution of these unsupported queries must be implemented at the application level by aggregating data through low level trace queries.

5.3 IBM Theseos

IBM's Theseos [11] is a query engine capable of processing complex queries across organizations to enable the development of traceability applications in a completely distributed setting. Theseos relies on a novel traceability data model that eliminates any data dependencies between organizations, which serves as a global schema that allows the formulation of a query without knowledge on how the data is stored or where it is located, and how a tracking query is executed [2]. In particular, Theseos introduces two attributes in its data model, namely `sentTo` and `receivedFrom`, that each organization is required to maintain for the movement path of an object. With this information, it is possible to minimize the number of nodes to be visited without flooding queries to all nodes in the network.

Traceability queries are first processed locally. Based on the outcome of this process, the query is further analyzed. It may be rewritten and then forwarded to other distributed databases. The results retrieved from the network are added to the local results and post-processed are required to yield the final response.

Figure 8 illustrates the role of the Theseos query engine to enable traceability queries. New information related to an object is captured, as a result of $\mathcal{O}$ moving from node $\mathcal{N}_1$ to $\mathcal{N}_2$ and then to $\mathcal{N}_3$. Other enterprise data such as the order request,

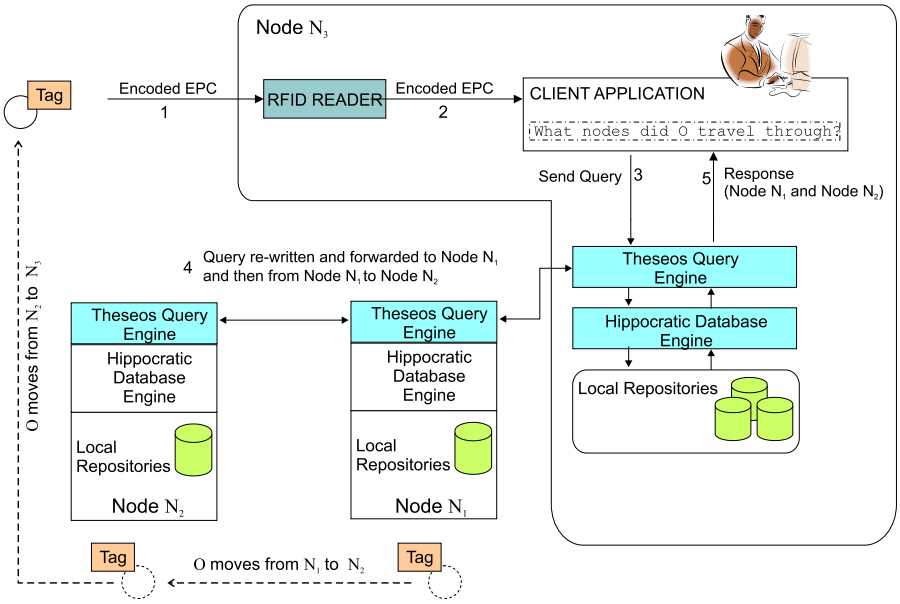


Fig. 8 IBM Theseos architecture example

dispatch records and billing information can be used to identify the source and destination of the object at each node to populate the `sendTo` and `receivedFrom` entries of the data model. When a client traceability application (at node $\mathcal{N}_3$ in this example) requires trace information regarding the object $\mathcal{O}$, a query is sent to the Theseos Query Engine. This query is processed locally to establish the next node to forward the trace query (in this example $\mathcal{N}_2$). Eventually the query reaches $\mathcal{N}_1$ and the query responses are gathered and processed at $\mathcal{N}_3$ and the object movement path $\{\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3\}$ is sent to the client application.

One advantage of Theseos is that the data is not centrally maintained and each organization has the ability to selectively share traceability data with other trading partners. Another advantage is its scalability. Since data is processed and stored in each individual node, the workload is naturally distributed.

Unfortunately, to obtain the `sentTo` and `receivedFrom` information, Theseos requires high synchronization with other enterprise data (e.g., billing or accounting information). This is impractical for many applications where such enterprise data may be unavailable. Another significant disadvantage of Theseos is its instability. If any of the peers is down, all queries relevant to that peer will fail. This is because of the difference between peer-based RFID solutions and other peer-based data sharing applications such as Bittorrent [60]. Bittorrent allows redundancy to exist and makes good use of this feature to increase data availability and reliability. But peer-based RFID solutions keep data strictly private at each node (i.e., there is no redundancy). More analysis on IBM Theseos are as the following:

Unique identifier. The data models of Theseos are capable of supporting any existing object identification mechanisms, including the EPC.

Uncertainty. This is not addressed in Theseos.

Prediction Query Support. This is also not addressed in Theseos.

Scalability. Theseos is a peer-to-peer based architecture that does not rely on services lookup such as ONS of DS. Consequently, it is a highly scalable solution to developing traceability applications.

Heterogeneity. Modules developed are application specific and do not have standardized interfaces or clear separation of roles as in the EPC Network. Theseos Query Engine depends on additional data such as billing information from Enterprise Resource Planning (ERP) systems. However, no standard mechanisms or interfaces are proposed to support the acquisition and utilization of such enterprise data.

Timely Responses. Theseos processes queries locally using available data and privacy policies. Then the queries are rewritten and sent to other nodes for processing. These actions are recursively repeated at nodes along the supply chain. The idea addresses the scalability issue of Discovery Services (see Sect. 5.2). However, response time to queries will be longer and unpredictable.

Privacy and Security. Theseos allows enterprises to selectively control access to traceability data using Hippocratic Database (HDB) technology based on ten principles rooted in privacy regulations [3]. Consequently, successful execution of a traceability query requires the inquiring party having the access privileges to the data stored at the nodes along the movement paths of an object.

Data Model. The two data model attributes of `sentTo` and `receivedFrom` are capable of realizing track and trace queries in a scalable architecture. This informa-

tion is obtained from enterprise data such as billing information for orders. The lack of a flexible and a generic approach to easily obtain such data poses difficulties in practice.

Traceability Queries. Theseos supports three specific application queries: (i) pedigree, (ii) product recall, and (iii) bill-of-materials. These queries are realized using trace queries (standard trace query and a containment trace query). However, prediction queries are not supported.

5.4 DIALOG

Distributed Information Architectures for cOllaborative loGistics (DIALOG) [15, 23] is a collection of projects focusing on developing a distributed information sharing system. The DIALOG system is an open-source solution built on a P2P architecture for tracking objects using the DIALOG middleware system illustrated in Fig. 9. Similarly to Theseos, DIALOG stores the data at places where it is collected. However, DIALOG requires the manufacturer of the tagged goods to maintain a server (software agent) which records the movement of the objects. Each *Information Provider* node notifies the specific DIALOG agent when an object is identified by its ID@URI identifier which consists of two components, a *unique ID string* and a *URI* (Uniform Resource Identifier) where the DIALOG Software Agent of the object resides.

The DIALOG System is a reportedly affordable solution to achieve a traceable network. DIALOG has the advantage that any organization can create a globally unique identifier in a decentralized manner at almost zero marginal cost (the URI could be based on a domain name that is already owned by the organization—or is cheap to register). This is in contrast to the EPC codes distributed by the EPCglobal to their subscribers who must pay a fee to be a member.

Most of the recent developments in DIALOG have been achieved through an EU funded project named PROMISE[17] (Product Lifecycle Management and Information Tracking using Smart Embedded Systems). A key development PROMISE made to

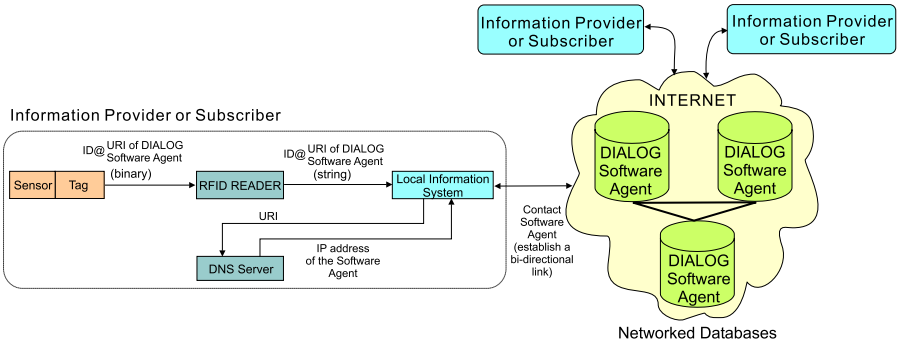


Fig. 9 The DIALOG architecture

[17] http://www.promise.no/.

DIALOG is the design and implementation of the Web service based PROMISE Messaging Interface (PMI). This interface enables the DIALOG architecture to support data models appropriate for capturing not only location information, but also other condition information such as those provided by embedded sensors [54].

Furthermore, the following extensions to DIALOG also sought to address some deficiencies in meeting the key technical requirement of timely responses:

Instance-level subscription. A new subscription mechanism in PMI implementation is capable of providing specific sensor readings, other information changes and events both through a call-back and a pull-based mechanism.

Time stamps. All messages handled by DIALOG (for instance events) have two timestamps: *event generation time* and *event reception time* at the current node. Timestamps allow unordered events to be correctly sequenced at the node where these events are processed. This helps DIALOG agents correctly interpret state information based on a sequence of events from different sources.

The central advantage of DIALOG lies in its open architecture where messages are automatically routed using existing DNS infrastructure. However, using DIALOG with the ID@URI product instance identifiers still does not address the issues of re-routing when URLs change and the relatively long identifiers of the DIALOG system. Consequently, the questions of what product instance identifiers should be used and how to lookup or discover information sources about those instances are still largely an open issue. The following summarizes suitability of DIALOG for building traceable RFID networks:

Unique identifier. DIALOG exploits the ID@URI unique object identifier. ID portion of the identifier can be used to encode a variety of other numbering schemes such as the EPC.

Uncertainty. This is not addressed in DIALOG.

Prediction Query Support. This is also not addressed.

Scalability. Although the software agents responsible for managing the information regarding objects may appear to be a bottleneck, DIALOG is a distributed system based on a robust multi-agent systems architecture that is able to mitigate such problems and actively manage work loads and connections to information providers and subscribers.

Heterogeneity. While DIALOG does not have specific standards, its software agents are capable of implementing existing standards such as EPCIS. The extensions to DIALOG made in the PROMISE enables DIALOG using PMI for data exchange. Similar to the EPC Network, DIALOG is also agnostic to the underlying identification technology.

Privacy and Security. Access control polices and authentication services implemented by the DIALOG software agents control parties that can send information to and access information from a DIALOG agent.

Data Model. The visible data model employed by DAILOG is currently based on PMI. Similar to the EPCIS event data model, PMI data models are flexible and extension can be made to suit application specific needs. Although PMI data models do not support temporal and spatial abstractions, and statistical constructs, PMI does support the explicit annotation of containment relationships and sensor data.

Traceability Queries. DIALOG supports both track and trace queries. However, prediction queries are not supported.

5.5 Hierarchical P2P-based RFID code resolution network

In a recent effort presented in [61, 62], the authors propose a hierarchical peer-to-peer (P2P) architecture to solve the load balancing and single node failure problems in ONS and DS type resolution infrastructure we have discussed in Sects. 5.1 and 5.2. As depicted in Fig. 10, the resolution infrastructure is divided into groups and each group has a super node and the collection of super nodes form a higher level group (i.e., super node group). All groups are organized as P2P networks with unique node identifiers. The super node of each group is usually maintained by the resolution service provider and the load in the group is distributed to the member nodes by its super node.

Figure 10 shows a two-layer architecture, but there can be more than two layers. The objects flowing through a supply chain network are assigned to groups according to the prefix of their EPC. The prefix selected is segmented into two portions C_1 and C_2 where C_1 is the manager number section of an EPC and C_2 segment corresponds to product type code and serial number portions of an EPC identifier. Group nodes maintain the mapping information between individual EPCs and services endpoints responsible for each EPC such as the URL of an EPCIS (see Group Node EPC Mapping Table in Fig. 10). When a query starts, super nodes transfer the query to the

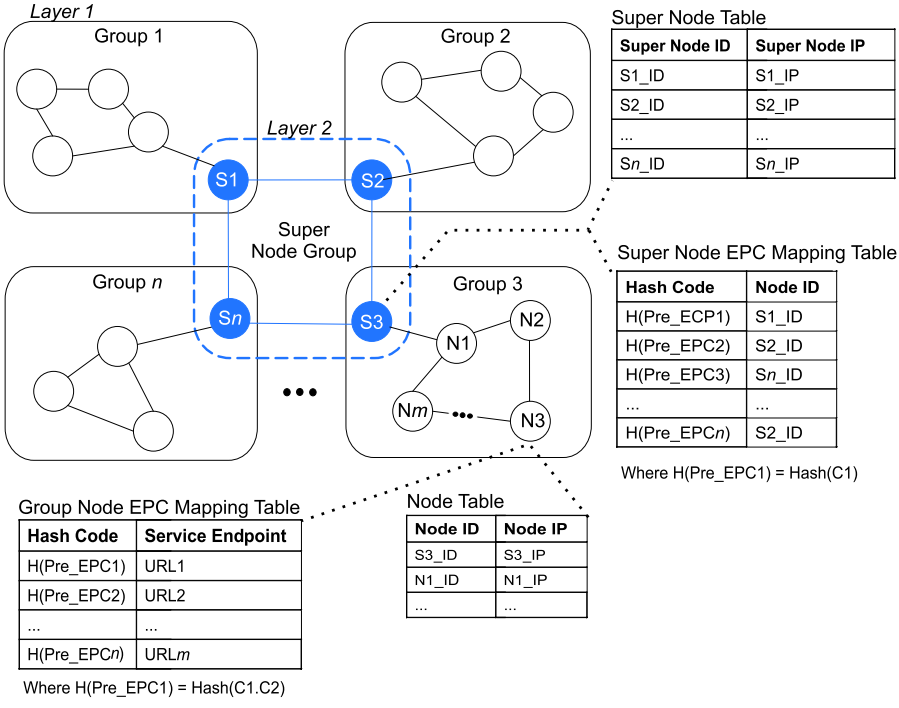


Fig. 10 The architecture of hierarchical P2P network

group nodes. Super nodes are responsible for both starting a query and returning the query result.

The proposed dynamic resolution network in [61] based on the static network described above is a hierarchical P2P solution for implementing a DS type infrastructure to record object movements and support track and trace type queries. Every event in a local EPC Network should be registered within the group nodes in the form of <EPC, EPCIS Address + event_type + timestamp>. Upon notification from a group node of the low level index, the super node of the group, S_i, creates a secondary index in the from of <EPC, S_i_IP_Address>. Finally, based on the P2P protocol used, one or more super nodes, for example S_k, are chosen by the group super node S_i to store the secondary index record in the form of <Hash(EPC), S_i_IP_Address>. It is then recorded in the secondary index table of super node S_k. Since each object has a unique EPC, all records related to a specific EPC generate the same secondary index hash values as the object moves through a supply chain network. Consequently, all the records related to a particular EPC is managed by a specific set of super nodes with pointers to group nodes that contain mappings between EPCs and service endpoints that observed the movement of the object. More analysis of this work are as the following:

Unique Identifier. The work is based on the EPC Network and unique identifiers of multiple formats are well supported.

Uncertainty. This is not addressed.

Prediction Query Support. This is also not supported.

Scalability. Similar to Theseos, this work is based on a pure P2P architecture that is highly scalable. The work load is also well balanced.

Heterogeneity. Static resolution network proposed for ONS benefits from an already ratified standard for accessing the ONS (refer to Fig. 4). Furthermore, current developments in DS will also see standardized DS interfaces such as DS Publish and DS Query Interfaces. Thus the proposed architecture will be able to ensure that traceable RFID networks are agnostic to the underlying implementations.

Timely Responses. Similar to Theseos, this P2P approach will result in longer and unexpectable response times.

Security and Privacy. The proposed architecture specifies a message authentication protocol to achieve three security and privacy objectives: (i) preventing the information leakage from the EPC, (ii) node authentication to ensure that all nodes in the resolution network are valid, and (iii) message validation to ensure the integrity of messages sent to and received from the resolution network.

Data Model. No specific models are provided to support high level abstractions or efficient processing of traceability queries.

Traceability Queries. The architectural extensions only provide a means to enable track and trace queries given an underlying EPC Network architecture as described in [61]. Hence there is no efficient mechanism for realizing high-level track and trace queries.

5.6 Evaluations and open issues

The aforementioned architectures are compared using the traceability requirements presented in Sect. 4. Table 1 summarizes the results. From the table we can see that

Table 1 Comparison: system architectures vs. system development requirements

	UID support	Scalability	Heterogeneity	Security and privacy	Timely response	Uncertainty	Prediction query support
EPCglobal Network	Provides a centrally managed EPC but requires subscription to EPCglobal	Distributed architecture. ONS infrastructure and the DS (in development) for references to distributed information sources (EPCIS repositories) per each unique object. Development of a security layer and a scalable architecture for DS are still undergoing	Vendor neutral and freely available standards. However the standardization of DS interfaces are still ongoing	Provided through authorization and access control services provided through EPCglobal to the ONS. Access control mechanisms defined for EPCIS services control entry to local EPC Networks	Supports real-time responses through subscription and call back mechanisms	Not supported	Not supported
BRIDGE	Uses EPC managed by EPCglobal	Similar to the EPCglobal Network but with DS implementations based on the Directory of Resources model [9]. DS may be a bottleneck due to complex security policies required to manage individual EPC records	Uses EPCglobal standards but new interfaces developed are yet to be standardized	Directory of Resources model and access control policies for individual EPC records ensure that any DS client will have to pass two access controls: (i) at the DS to obtain the list of EPCIS links and (ii) at the subsequent query to the EPCIS to obtain detailed event information)	Subscribing mechanism for real-time responses	Partially addressed through probabilistic algorithms and supply chain network data model. Algorithms are based on hidden Markov models (HMM) and are available to support applications through APIs	Partially addressed through a supply chain network data model and HMM
IBM Theseos	Able to use existing UIDs including EPC	Distributed and P2P based architecture, highly scalable	Vendor neutral standards based on EPCglobal standards	A peer controls access to its own data using fine grained access controls based on Hippocratic DB technology	Cannot guarantee real-time responses because of P2P query latency	Not addressed explicitly	Not addressed explicitly

Table 1 (*Continued*)

	UID support	Scalability	Heterogeneity	Security and privacy	Timely response	Uncertainty	Prediction query support
DIALOG	Uses the ID@URI identifier. Relies on each company owning a unique domain name (however costs are minimal)	Multi-agent based distributed system. Single agent for a product class or type may be a bottleneck	No DIALOG specific standards. However, DIALOG agents support implementation of any data exchange standard such as PMI in PROMISE. Although ID@URI approach is flexible, URLs in the identifier are fragile and inflexible	Data source (e.g., manufacturers) nominated by ID@URI have control over all the data about objects and partner nodes need to update the data source. Other participants therefore lack control on the data	Extensions made to support real-time response through call-back and pull-based mechanisms	Not addressed explicitly	Not addressed explicitly
Hierarchical P2P Network	Uses EPC managed by EPCglobal	Distributed and P2P based, highly scalable	Uses vendor neutral standards developed by EPCglobal	Defined security mechanism for communications between peers and access to data	Real-time responses cannot be guaranteed because of P2P query latency	Not addressed explicitly	Not addressed explicitly

despite recent progress in RFID traceability research, many issues still remain to be resolved at the system architecture level in RFID traceability networks.

5.6.1 Unique identifier

Tracking, tracing and predicting the state of individual objects require an unique identification scheme that can be manipulated by information systems. EPCglobal Network, BRIDGE and the Hierarchical P2P based architectures are all built upon EPC. DIALOG relies on ID@URI approach. The cost of guaranteeing the uniqueness of the EPC's company identifier (manager number) requires subscription to EPCglobal.[18] ID@URI approach relies on each company owning a unique domain name.

There are three significant differences between the two approaches: (i) the cost of domain name registration is nominal while subscription to EPCglobal is more expensive, (ii) as a result of using URIs, the DIALOG architecture requires relatively more expensive RFID tags (rewritable) compared to write-once RFID tags used with the EPCglobal approach, and (iii) EPCs are centrally managed by EPCglobal, which guarantees the uniqueness with a global scope. The URI used in the DIALOG system is usually a URL, which is quite fragile. For example, if the URL or more specifically the local path of the software agent is changed, objects whose tags have been written with the URL might fail to resolve on the DIALOG network.

5.6.2 Uncertainty and prediction query support

Most of the existing architectures do not address uncertainty explicitly. The assumption that the underlying data capturing technologies are perfect is not correct. Uncertainty in captured data significantly affects the results generated by traceability queries. In recent years, uncertainty has become an active research topic [1, 10].

Only architecture extension supported by BRIDGE has provided high-level models and algorithms capable of modeling the uncertainty. However, an important issue with the approach is the need of supervised learning for the models to be useful. Significant work is needed to (i) investigate other modeling techniques such as conditional random field (CRF), skip chain CRF [55] and Emerging Patterns [30] and (ii) consider more dynamic models that does not require a learning phase [31, 50].

5.6.3 Scalability

The existing architectures have achieved scalability based on either federated or P2P architectures. EPCglobal Network and BRIDGE are federated. The problem with this approach is that the Discovery Service (Serial Lookup Service in BRIDGE) becomes a bottleneck. This issue is considered by the Hierarchical P2P approach by implementing Discovery Service in a pure distributed manner (refer to Sect. 5.5) and by DIALOG (refer to Sect. 5.4) based on its multi-agent design.

[18] No subscription is required for use of USDOD-64 and USDOD-96 EPC identifier types designated for North American military use.

In general, in a P2P based approach, each node delegate a query to its neighbors if it cannot answer the query itself. For example, in Theseos, queries are processed locally and re-written before forwarded to the next node. However, a significant issue for P2P based approach is that each node must take an equal equity in the network and be open to the idea of having its data stored on different peers that may be controlled by competitive businesses.

5.6.4 Heterogeneity

Most of the existing architectures follow the EPCIS standards in the EPCglobal Architecture Framework so that the backend system implementations do not affect the high-level architecture. Perhaps one of the most important distinctions is that EPCglobal Network provides a layered architecture stack with well defined standard interfaces. In contrast, DIALOG does not provide standard interfaces such as the EPCIS for exchanging object related data.

Use of standards is critical when dealing with data exchange between multiple organizations and heterogeneous environments. Standards also play a role in enhancing competition in the technology and system provider market by allowing multiple vendors to compete for system components and technologies.

5.6.5 Timely response

EPC Network, BRIDGE and DIALOG use subscription mechanisms ("push" based approaches) to support real-time data requirements. However in query processing, P2P architectures are not capable of providing time constraints although this is an important requirement for traceability applications. For P2P architectures, a query may be propagated several times and this significantly increases the processing time.

Improving the timeliness of query responses is a significant issue and we encourage more research in this direction.

5.6.6 Security and privacy

To enable traceability in a distributed RFID system, there must be some level of data sharing between nodes. Access control in DIALOG is shifted away from parties down the supply chain towards manufacturers. Manufacturers of objects exercise dominant control over collecting information from other parties and sharing of that information with client applications. In contrast, the EPCglobal Network (through the Discovery Services mechanism developed in BRIDGE) allows highly granular access control policies to be specified by parties collecting information in order to determine access rights by other entities to product related data.

There are trade-offs between the privacy and data sharing. For example, P2P architectures protect privacy adequately by providing nodes with ownership of the data and the choice to respond only queries from desirable parties, which also means some constraints for data sharing. Thus techniques for dealing with the tradeoff between privacy and data sharing is still a research challenge.

6 State-of-the-art RFID traceability data models

Data models determine the structure of low level data storage and representation. Designing appropriate data models significantly affects the performance of the whole system. Due to the nature of large-scale traceability applications (e.g., high volume of data, distributed across organizations, complicated relationship such as containment), data models must be appropriately designed. Fortunately, database research community is recently developing a strong interest in RFID data modeling [2, 11, 13, 25, 26, 33, 39, 40, 56, 57, 61]. In this section, we will examine a set of representative research work on data modeling and corresponding query processing techniques for RFID traceability networks. We analyze each work by considering the data model requirements identified in Sect. 4.2. For each data model, we also analyze its query processing performance by examining the following three typical traceability queries:

Track Query. Returns the current status of an object (Q1 in Sect. 3.2).
Trace Query. Trace queries are used to discover the history of an item. We use Q2 in Sect. 3.2 in the evaluation.
Statistical Trace Queries. Returns summary statistical information based on the historical status of objects. We use Q15 in Sect. 3.2 in the evaluation.

It should be noted that other queries also can be used in the evaluation. In the performance analysis, we assume there is no additional indices except the primary key and the indices are all *b*-order B+ Trees.

6.1 DRER model

Dynamic Relationship ER (DRER) [56] is the data model used by Siemens' RFID middleware system (Fig. 11). It abstracts static and dynamic entities including *object*, *reader*, *location* and *transaction*. Interactions are modeled as either state or event based relationships. The data model also provides a rule-based data filter engine. In [56], temporal and spatial tracing queries are considered, but statistical queries are

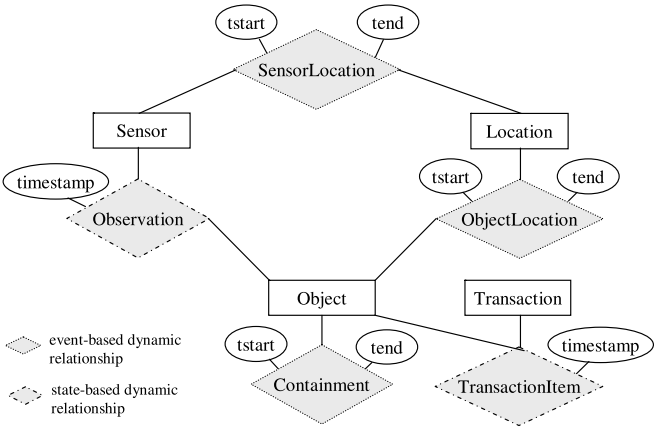


Fig. 11 DRER model

not covered. Although the design takes containment into consideration by introducing the *Containment* table, it does not mention how this relationship is captured.

DRER is one of first data models for managing large RFID databases. It models the transition of states for objects using the dynamic relations (e.g., *OBJECTLOCATION*). This model is simple yet expressive that it can be used to answer all the queries that we summarized in Sect. 3.2, either directly or by composing the low-level queries. Since DRER does not store any redundant information, it is efficient in storage. The problems of the DRER data model are:

- The model assumes that all relevant data are stored in a single database. It is hard to be used in a distributed system.
- The model focuses on modeling RFID events and does not take all traceability queries into account (e.g., path queries).
- The model lacks support for statistical queries.

6.1.1 Performance analysis

Track query The following query[19] is used to answer the track query Q1.

```
SELECT TOP 1 tend, location_id
FROM OBJECTLOCATION
WHERE epc = o
ORDER BY tend DESC
```

The main cost of this query is to find all records for the given *epc* using the *OBJECTLOCATION* table's index on *epc*, which takes $O(\log_b (n * m))$ in the worst case where n is the total number of objects, m is the average number of nodes visited by an object, and b is the order of B+ Trees. These records are sorted by *tend*. The sorting can be done in memory because the number of records is small (an object only appears at, at most all the locations, and the number of locations is usually small), which can be ignored.

Trace query To answer Q2, we can use:

```
SELECT location_id
FROM OBJECTLOCATION
WHERE epc = o
AND tend <= (
        SELECT min(tend) FROM OBJECTLOCATION
        WHERE epc = o and location_id = x)
ORDER BY tend ASC
```

Although an embedded query is used, a clever database engine will implement the inner query as an on-the-fly filter on the results of the outer query. So the cost is the same as track queries, i.e., $O(\log_b (n * m))$.

[19] For the convenience of discussion, we use MySQL-specific SQL.

Statistical query To answer Q15, a *count* function is performed on top of a self-join:

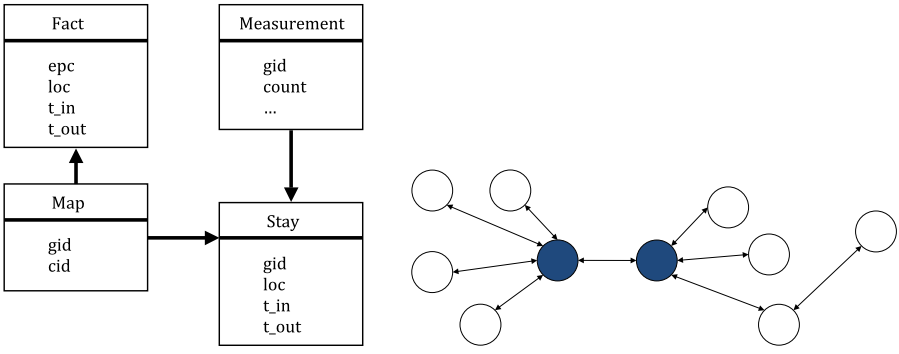
```
SELECT count(*)
FROM OBJECTLOCATION ol1
INNER JOIN OBJECTLOCATION ol2
ON ol1.epc = ol2.epc
AND ol1.location_id = x AND ol2.location_id = y
AND ol1.tend < ol2.tstart
```

Because time period *tend* and *tstart* are involved, this query cannot be answered using only index. Suppose the join operation is implemented as the sort-merge algorithm, the cost of this query is $O((n * m)/p)$ I/O operations, where p is the page size. This is much higher than the other two kinds of queries. We can conclude that *DRER* supports both track and trace queries very well. However, it lacks support for statistical queries. To answer these queries, one has to use complicated and inefficient low-level queries.

6.2 RFID-cuboid

The idea of RFID-Cuboid [26] is based on the observation that individual objects tend to move and stay together (i.e., bulky object movements). The records for the objects moving along the same segments can be merged without loss of information. The term "cuboid" implies that data is merged at some point. Compared with DRER, RFID-Cuboid is a static data mining model instead of dynamic event-driven model. Figure 12a shows the essential tables for the RFID-Cuboid. The *Fact* table is exactly the same as the *OBJECTLOCATION* table in DRER. RFID-Cuboid introduces the *Stay* and *Map* tables for data compression and measurement.

The most important advantage of RFID-cuboid is the efficient support of statistical and path-oriented queries, by grouping the objects and materialization of the group information. This materialization significantly improves the performance of query processing. However, the storage used by RFID-Cuboid is more than that by DRER because of the additional tables. This additional storage cost is further reduced by



(a) Structure of RFID-Cuboid (b) Gateway-based Movement Graph Model

Fig. 12 RFID-cuboid

the same authors' recent work by introducing a *Gateway-Based Movement Graph* model [25]. This enhanced work makes an assumption that there are some "gateway" nodes in an RFID network, which have either high fan-in or high fan-out edges as illustrated in Fig. 12b. These gateway nodes connect the sub-graphs together.

The RFID-Cuboid can be established around the gateway nodes, and de-grouped within the sub-graphs. Instead of using the starting location of a group of objects as the root, the gateway-based movement graph selects the gateway nodes as the root. In this way, the root groups are the largest, so that the number of groups are minimum. As a result, the size of *Stay* and *Map* tables are further reduced. This is very useful for large-scale, distributed traceability applications (e.g., global supply chain systems).

The RFID-Cuboid model efficiently compresses the data and improves the performance of queries using a tree-structure. However, it is highly dependent on the data distribution. The performance is significantly affected if objects do not exhibit bulky movements. Consequently, this data model is only suitable for large datasets that share some common properties (e.g., move together in bulky mode). The issues of RFID-Cuboid are summarized as the following:

- Similar to the DRER model, RFID-Cuboid assumes a centralized database.
- The data model does not capture containment relationships. *Stay* and *Map* tables need to be carefully maintained to ensure consistency of data.
- The performance quickly degrades when the objects do not move in groups, or even when they move in very small groups.

6.2.1 Performance analysis

Track query Because the *Fact* table is the same as the *OBJECTLOCATION* table in DRER, the track queries can be answered in the same way. On the other hand, the new tables do not help with item-level queries. So the performance is the same as that of DRER. However, in an environment where objects move in groups, RFID-Cuboid can answer the track by directly querying the *Stay* table when the input is a group.

Trace query The analysis is same as track queries discussed above.

Statistical query Answering statistical query is easy and fast with RFID-Cuboid. Instead of querying the fact table, we can query the *Stay* table as follows:

```
SELECT SUM(m1.count)
FROM ((
          SELECT m.cid AS id FROM Stay s
          INNER JOIN Map m ON s.gid = m.gid
          WHERE loc = l1
          )
          INTERSECT
          (
          SELECT gid AS id FROM Stay s
          WHERE loc = l2
          ) AS ids)
INNER JOIN Measurement m1 ON m1.gid = ids.id;
```

The strategy is to find the groups that moved from location l_1 to l_2 by intersecting the *Stay* records for both locations. RFID-Cuboid assigns the IDs for the subgroups by appending a unique number to their parent group. In this way, the movement direction is naturally inferred from the IDs. All the selection and join are performed on the *Stay* and *Map* table. With the assumption that objects move in groups, *Stay* table is much smaller than the *Fact* table. Thus, the query cost is much smaller. *Measurement* is the materialization for the groups.

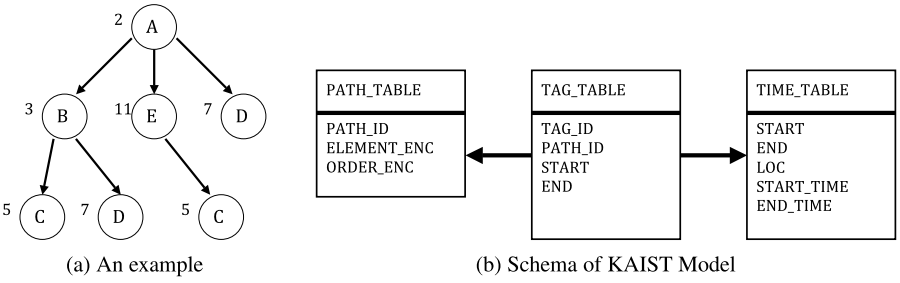
6.3 KAIST trace model

The researchers from KAIST (Korea Advanced Institute of Science and Technology) proposed a novel model to efficiently encode and query path information in an RFID database [39]. The encoding scheme is based on the Chinese Remainder Theorem and can encode a path to a serial number level. A query processing language is also proposed. The idea of this model is to represent the paths as a forest. Each starting location is presented by the root of a tree and receiving locations are child nodes of sending locations. Figure 13a shows an example.

Each node in the trees are marked by a unique prime number. Thus, each path can be encoded by a number namely *Element List Encoding Number* (*ELEMENT_ENC*), which is the multiplication of the prime numbers from the root to the leaf node. For example, the path $A \rightarrow B \rightarrow C$ is encoded as $2 * 3 * 5 = 30$. Moreover, according to *Chinese Remainder Theorem*, an *Order Encoding Number* (*ORDER_ENC*) is introduced to calculate the order of a node in a path. Suppose the prime number for a node x is $\mathcal{P}_x$ and the order of the node is o_x, we have $o_x = ORDER\_ENC \ mod \ \mathcal{P}_x$. A schema is proposed on top of the encoding method as shown in Fig. 13b. Interested readers are referred to [39] for more details.

Experiments have proved that this encoding scheme with the query processing method efficiently discovers the path information for a given object. In particular, for most queries, the KAIST model is better than RFID-Cuboid. In addition, similar to the RFID-Cuboid model, it significantly decreases the data storage size. However, this model does not assume that the object moves in groups, so it can be used in more scenarios. The KAIST model is path-centric and can efficiently process path-oriented queries. The problems of this model are summarized as the following:

- The encoding scheme cannot be used in a distributed environment. It does not represent containment relationship so cannot support corresponding queries.



(a) An example (b) Schema of KAIST Model

Fig. 13 KAIST model

- The model stores time information in a separate table, which introduces join operations for any query involving time conditions.

6.3.1 Performance analysis

Track query To answer track query Q1, first we join the *TAG_TABLE* and *PATH_TABLE* to get the *ELEMENT_ ENC* and *ORDER_ENC* for the object. Although this is a join operation, the predicate for *TAG_TABLE* (T.TAG_ID=x) makes sure only one record returned. Using hash-join algorithm, this can be done with $O(\log_b n + \log_b m')$ where m' is the size of the *PATH_TABLE* table.

Trace query The process to answer Q2 is the same as that for track query except for the last step where we return the list of nodes sorted by their orders, instead of only returning the one with largest order.

Statistical query The following query can be used to answer query Q15.

```
SELECT count(*)
FROM PATH_TABLE p INNER JOIN TAG_TABLE t
ON p.ELEMENT_ENC mod Px * Py
AND p.ORDER_ENC mod Py - p.ORDER_ENC mod Px = 1
AND p.PATH_ID = t.PATH_ID
```

The statistical query processing is similar to that of DRER model. They both perform a join on two tables. The difference is that the *PATH_TABLE* table in KAIST is much smaller than *TAG_TABLE*. Since the size of *TAG_TABLE* is almost the same as *OBJECTLOCATION* table in DRER, the query processing cost is smaller than DRER. However, because the time information is stored in a separate table, another join operation has to be performed if there is any time conditions in queries.

6.4 SPIRE

None of the aforementioned models discuss how the containment relationship is captured. How to automatically infer the containment relationship is still an open problem. It is a multiple-layer problem which involves hardware configuration, data cleansing, uncertain data management and other techniques. *SPIRE* is a representative work on this problem.

An earlier work from SPIRE [12] proposes two options to detect containment relationships. One option is the manual approach, the other is to configure RFID readers so that it can read only the most outer container's tag. However, it does not solve the regrouping problem. In addition, due to the unreliability of RFID readers, it will not work perfectly even with the assumption that no regrouping exists. In a recent work by the same authors [13], the SPIRE system is improved to detect containment relationship using a statistical method. In this approach, the containment is inferred by the historical co-location of tags. A time-varying colored graph model is proposed as shown in Fig. 14.

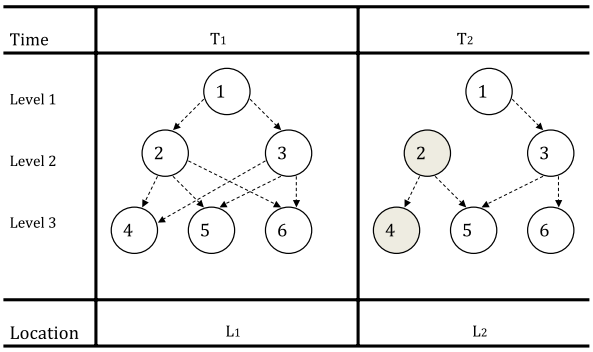


Fig. 14 The SPIRE model

The edges indicate *possible* containment relationship, while the objects detected together at the same location are marked with the same color. At the beginning, the edges are added from the higher level container to lower level objects/containers if they are at the same location. After they move to a new location, some edges are removed if the co-location relationship does not exhibit any more. Ideally, after some point, there should be exactly one path from the root to a certain leaf. However, this rarely happens because of regrouping. To solve this problem, a probabilistic inference method is proposed in SPIRE. The basic idea is to assign weights to the co-location records and recent record gets higher weight. The incoming edges to a node are sorted by the weighted sums of the co-location records. SPIRE chooses the edge with highest sum to update the containment relationship.

This model partially solves the containment relationship detection problem. However, it suffers from several disadvantages:

- The model depends on a kind of "special reader" and the packaging level information in the tag data to confirm the containment relationships.
- The containment relationship is inferred by co-location of tags, and specifically, only with changes to the co-location relationship.
- The algorithm is probabilistic and the result is not accurate. SPIRE does not mention how to model the inaccuracy of inferences.

6.4.1 Performance analysis

SPIRE focuses on how to automatically infer the containment relationship. It does not discuss how to process regular traceability queries. The containment detection algorithm itself is $O(n_0 * n_1 * \cdots * n_k)$, where n_i is the total number of records at level i for the worst case (i.e., when all the objects move together all the time).

6.5 Evaluations and open issues

Underlying data models play an important role in shaping the higher level architectures for RFID traceability networks. A well-designed data model can significantly improve system performance and decrease persistent data storage requirements. We compare the aforementioned data models using the requirements identified in Sect. 4.2, as well as the traceability queries supported by each data model. Tables 2

Table 2 Comparison: data models vs. data model requirements

	Temporal abstractions	Spatial abstractions	Statistical constructs	Containment relationships	Uncertainty
DRER	Transaction entity	Location entity	Not addressed explicitly	Observation event	Not addressed
RFID-Cuboid	Stay table	Stay table	Data mining queries	Not addressed explicitly	Not addressed
KAIST	Time table	Path table and path encoding scheme	Partially addressed	Not addressed	Not addressed
SPIRE	Time stamps	Object-Containment Graph	Not mentioned	Object-Containment Graph	Not addressed

Table 3 Comparison: data models vs. supporting traceability queries

	Trace	Trace (temporal constraints)	Trace (spatial constraints)	Trace (statistical)	Trace (containment)
DRER	Partially supported (path related queries not supported)	All supported	All support	Not supported directly[a]	Partially supported
RFID-Cuboid	All supported	All supported	All supported	Supported	No
KAIST	All supported	All supported	All supported	Not supported directly	No
SPIRE	Not supported directly	All supported	All supported	Not supported directly	Supported

[a]"Not supported directly" implies that these queries can only be answered by composing low level queries

and 3 summarize the results. From the tables we can see that significant work remains in RFID data models and traceability query processing:

Distributed data model. Most RFID traceability applications are distributed and spread across organizations. It is difficult to assume or require data to be stored in centralized databases. Distributed data models therefore need to be carefully designed to support traceability queries. However, at the time of writing, to the best of our knowledge, there are no existing data models that meet all the requirements we have outlined. We believe that extensive researches are needed for modeling distributed RFID data.

Statistical queries over paths. Realizing these queries can provide data flow statistics through particular nodes or paths, which is vital for high-level business decisions in traceability applications. Unfortunately, these statistical queries are not well supported by existing data models.

Table 4 Comparison: data models vs. performance

	Track query	Trace query	Trace (statistical) query	Storage efficiency
DRER	$O(\log_b n * m)$	$O(\log_b n * m)$	$O(n * n/p)$	$O(n)$
RFID-Cuboid	$O(\log_b n * m)$	$O(\log_b n * m)$	$O(n' * n'/p)$	$O(n + n')$
KAIST	$O(\log_b n + \log_b m')$	$O(\log_b n + \log_b m')$	$O(n * m'/p)$	$O(n + m')$
SPIRE	N/A	N/A	N/A	N/A

n is the number of objects. m is the average number of locations visited by an object. n' is the size of *Stay* table. m' is the size of *PATH_TABLE* table. p is the page size

Containment queries. Containment queries over object paths are also important, especially in product recalls where an object (e.g., tainted pork) from a node should be recalled. For these scenarios, it is necessary to find all other objects (e.g., other pork that traveled in the same pallet) that have a containment relationship with the object in question, obtain their paths and recall them. Unfortunately, containment queries are also not supported by most of the existing data models.

Uncertainty. Most existing data models do not take uncertainty of RFID data into consideration. However, as we have discussed in Sect. 4, uncertainty should be treated as first class citizen in RFID traceability networks.

We also compare the data models for their time and space efficiency. The results are shown in Table 4. We can see that the DRER model is efficient in storage but in general the KAIST model outperforms others in query processing. However, as discussed in the performance analysis, all proposed data models have some disadvantages and issues. Future research is needed to find a model that is not only expressive, but also having good performance on storage and query processing.

7 Conclusion

During the past decade, RFID technologies have developed rapidly and are increasingly used in large-scale, mainstream applications. Traceability is a critical aspect of majority of these RFID applications. Enabling traceability using networked RFID systems brings some fundamental research and development issues. Most of these challenges are due to the large volume of data generated from different organizations, the unwillingness of participants to share data, and data quality issues that arise as a result of the physical layer. Consequently, we need to propose novel solutions to make RFID traceability applications scalable, robust, and secure. In particular, a well-defined data model that takes into consideration unique characteristics of RFID traceability applications such as temporal and spatial characteristics of RFID data, uncertainty and containment relationships that exist between objects.

We surveyed the current approaches to enable traceability in RFID networks. By comparing and analyzing these architectures and data models, we have concluded that the area of RFID traceability networks presents many interesting challenges and opportunities that need to be resolved before global traceable RFID networks can be

fully realized. Along with current research and development efforts, we encourage more insight into the problems of RFID traceability networks, and more innovative solutions to the open research issues identified in this paper.

Acknowledgements This research has been supported by the Australian Research Council Discovery Grant DP0878917.

References

1. Agrawal, P., Benjelloun, O., Sarma, A.D., Hayworth, C., Nabar, S., Sugihara, T., Widom, J.: Trio: a system for data, uncertainty, and lineage. In: Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB'06), Seoul, Korea (2006)
2. Agrawal, R., Cheung, A., Kailing, K., Schonauer, S.: Towards traceability across sovereign, distributed RFID databases. In: Proceedings of the 10th International Database Engineering and Applications Symposium (IDEAS'06), Delhi, India (2006)
3. Agrawal, R., Kiernan, J., Srikant, R., Xu, Y.: Hippocratic databases. In: Proceedings of the 28th International Conference on Very Large Data Bases (VLDB'02) (2002)
4. Ahmed, N., Ramachandran, U.: RFID middleware systems: a comparative analysis. In: Unique Radio Innovation for the 21st Century: Building Scalable and Global RFID Networks. Springer, Berlin (2010)
5. Aigner, M., Feldhofer, M.: Secure symmetric authentication for RFID tags. In: Proceedings of the Telecommunication and Mobile Computing (TCMC'05), Graz, Austria (2005)
6. Alexander, K., Gilliam, T., Gramling, K., Grubelic, C.: Applying auto-ID to reduce losses associated with shrink. http://www.autoidlabs.org/uploads/media/IBM-AUTOID-BC-003.pdf
7. Angeles, R.: RFID technologies: supply-chain applications and implementation issues. Inf. Syst. Manag. **22**(1), 51–65 (2005)
8. Buckley, L.M., Olson, C.W.: High tech, high stakes: using technology to smash the fake trade. In: IPWorld, pp. 30–33 (2005)
9. Cantero, J.J., Guijarro, M.A., Plaza, A., Arrebola, G., Baños, J.: A design for secure discovery services in the EPCglobal architecture. In: Unique Radio Innovation for the 21st Century: Building Scalable and Global RFID Networks. Springer, Berlin (2010)
10. Cheng, R., Singh, S., Prabhakar, S.: U-DBMS: a database system for managing constantly-evolving data. In: Proceedings of the 31st International Conference on Very Large Data Bases (VLDB'05) (2005)
11. Cheung, A., Kailing, K., Schönauer, S.: Theseos: a query engine for traceability across sovereign, distributed RFID databases. In: Proceedings of the 23rd International Conference on Data Engineering (ICDE'07), Istanbul, Turkey (2007)
12. Cocci, R.: SPIRE: scalable processing of RFID event stream. In: Proceedings of the 5th RFID Academic Convocation, Brussels, Belgium (2007)
13. Cocci, R., Tran, T., Diao, Y., Shenoy, P.: Efficient data interpretation and compression over RFID streams. In: Proceedings of the 24th International Conference on Data Engineering (ICDE'08), Cancun, Mexico (2008)
14. DeHoratius, N., Raman, A., Ton, Z.: Execution the missing link in retail operations. Calif. Manag. Rev. **43**(3), 136–151 (2001)
15. DIALOG: Distributed Information Architectures for cOllaborative loGistics. http://dialog.hut.fi/
16. Dimitriou, T.: A lightweight RFID protocol to protect against traceability and cloning attacks. In: Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05), Athens, Greece (2005)
17. EPCGLOBAL: EPCGLOBAL. http://www.EPCGLOBAL.com
18. EPCglobal: EPCglobal Specifications. http://www.epcglobalinc.org/standards/specs/
19. europa.eu: General Food Law—Traceability. http://ec.europa.eu/food/food/foodlaw/traceability/
20. FDA: Combating counterfeit drugs, a report of the food and drug administration. http://www.fda.gov/oc/initiatives/counterfeit/report02_04.html
21. Feldhofer, M., Dominikus, S., Wolkerstorfer, J.: Strong authentication for RFID systems using the AES algorithm. In: Proceedings of the 6th International Workshop on Cryptographic Hardware and Embedded Systems (CHES'04), Cambridge, USA (2004)

22. Finkenzeller, K.: RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification. Wiley, New York (2003)
23. Främling, K., Nyman, J.: From tracking with RFID to intelligent products. In: Proceedings of 14th IEEE International Conference on Emerging Technologies and Factory Automation, Palma de Mallorca, Spain (2009)
24. Globeranger: Globeranger. http://www.Globeranger.com
25. Gonzalez, H., Han, J., Cheng, H., Li, X., Klabjan, D., Wu, T.: Modeling massive RFID data sets: a gateway-based movement graph approach. IEEE Trans. Knowl. Data Eng. **22**, 90–104 (2010)
26. Gonzalez, H., Han, J., Li, X., Klabjan, D.: Warehousing analyzing massive RFID data sets. In: Proceedings of the 22nd International Conference on Data Engineering (ICDE'06), Atlanta, USA (2006)
27. Gotel, O.C.Z., Finkelstein, A.C.W.: An analysis of the requirements traceability problem. In: Proceedings of the 1st International Conference on Requirements Engineering (ICRE'94), Colorado Springs, CO, USA (1994)
28. California Government: California business and professions code sections 4163. http://www.leginfo.ca.gov/calaw.html
29. GS1: GS1 Traceability. http://www.gs1.org/productssolutions/traceability
30. Gu, T., Wu, Z., Tao, X., Pung, H.K., Lu, J.: epSICAR: an emerging patterns based approach to sequential, interleaved and concurrent activity recognition. In: IEEE International Conference on Pervasive Computing and Communications, Los Alamitos, CA, USA (2009)
31. Huynh, T., Fritz, M., Schiele, B.: Discovery of activity patterns using topic models. In: Proceedings of the 10th International Conference on Ubiquitous Computing (Ubicomp '08), Seoul, South Korea (2008)
32. Ilic, A., Andersen, T., Michahelles, F.: Increasing supply-chain visibility with rule-based RFID data analysis. IEEE Internet Comput. **13**(1), 31–38 (2009)
33. Jeffery, S.R., Garofalakis, M., Franklin, M.J.: Adaptive cleaning for RFID data streams. In: Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB'06), Seoul, Korea (2006)
34. Juels, A.: RFID security and privacy: a research survey. IEEE J. Sel. Areas Commun. **24**(2), 381–394 (2006)
35. Juels, A., Pappu, R.: Squealing Euros: privacy protection in RFID-enabled banknotes. In: Financial Cryptography, pp. 103–121. Springer, Berlin (2002)
36. Kelepouris, T., Baynham, T., McFarlane, D.: Track and trace case studies report. http://www.autoidlabs.org/uploads/media/AUTOIDLABS-WP-BIZAPP-035.pdf, 2006
37. Ketzenberg, M., Ferguson, M.: Managing slow moving perishables in the grocery industry. Prod. Oper. Manag. **17**(5), 513–521 (2008)
38. Landt, J.: The history of RFID. IEEE Potentials **24**(4), 8–11 (2005)
39. Lee, C.-H., Chung, C.-W.: Efficient storage scheme and query processing for supply chain management using RFID. In: Proceedings of the 28th ACM SIGMOD International Conference on Management of Data (SIGMOD'08), Vancouver, Canada (2008)
40. Lin, D., Elmongui, H.G., Bertino, E., Ooi, B.C.: Data management in RFID applications. In: Database and Expert Systems Applications. Lecture Notes in Computer Science, vol. 4653, pp. 434–444 (2007)
41. Lopez, T.S., Ranasinghe, D.C., Patkai, B., McFarlane, D.: Taxonomy technology and applications of smart objects. Inf. Syst. Front. **13**(2), 281–300 (2011)
42. Osaka, K., Takagi, T., Yamazaki, K., Takahashi, O.: An efficient and secure RFID security method with ownership transfer. In: RFID Security, pp. 147–176. Springer, New York (2009)
43. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-oriented computing: state of the art and research challenges. IEEE Computer **40**(11), 38–45 (2007)
44. Ranasinghe, D.C., Cole, P.H.: Networked RFID Systems and Lightweight Cryptography: Raising Barriers to Product Counterfeiting. Springer, Berlin (2008)
45. Ranasinghe, D.C., Harrison, M., Främling, K., McFarlane, D.: Enabling through life product-instance management: solutions and challenges. J. Netw. Comput. Appl. **34**(3) (2011)
46. Ranasinghe, D.C., Sheng, Q.Z., Zeadally, S.: Unique Radio Innovation for the 21st Century: Building Scalable and Global RFID Networks. Springer, Berlin (2010)
47. Rantzau, R., Kailing, K., Beier, S., Grandison, T.: Discovery services—enabling RFID traceability in EPCglobal networks. In: 13th International Conference on Management of Data (COMAD'06), Delhi, India (2006)
48. Rieback, M.R., Crispo, B., Tanenbaum, A.S.: The evolution of RFID security. IEEE Pervasive Comput. **5**(1), 62–69 (2006)

49. Robson, C., Watanabe, Y., Numao, M.: Parts traceability for manufacturers. In: Proceedings of the 23rd International Conference on Data Engineering (ICDE'07), Istanbul, Turkey (2007)
50. Rosen-Zvi, M., Chemudugunta, C., Griffiths, T., Smyth, P., Steyvers, M.: Learning author-topic models from text corpora. ACM Trans. Inf. Sys. **28**, 4:1–4:38 (2010)
51. Roussos, G.: Networked RFID: Systems, Software and Services. Springer, Berlin (2008)
52. Sheng, Q.Z., Li, X., Zeadally, S.: Enabling next-generation RFID applications: solutions and challenges. IEEE Computer **41**(9), 21–28 (2008)
53. Sheng, Q.Z., Wu, Y., Ranasinghe, D.C.: Enabling scalable RFID traceability networks. In: Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Application, Perth, Australia (2010)
54. Stark, J.: Product Lifecycle Management: 21st Century Paradigm for Product Realisation. Springer, Berlin (2004)
55. Sutton, C., McCallum, A.: An introduction to conditional random fields for relational learning. In: Getoor, L., Taskar, B. (eds.) Introduction to Statistical Relational Learning. MIT Press, Cambridge (2007)
56. Wang, F., Liu, P.: Temporal management of RFID data. In: Proceedings of the 31st International Conference on Very Large Data Bases (VLDB'05), Trondheim, Norway (2005)
57. Wang, F., Liu, S., Liu, P.: Complex RFID event processing. VLDB J. **18**(4), 913–931 (2009)
58. Wang, F., Liu, S., Liu, P.: A temporal RFID data model for querying physical objects. Pervasive Mob. Comput. **6**(3), 382–397 (2010)
59. Want, R.: An introduction to RFID technology. IEEE Pervasive Comput. **5**(1), 25–33 (2006)
60. Wei, B., Fedak, G., Cappello, F.: Scheduling independent tasks sharing large data distributed with BitTorrent. In: Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing (GRID'05), Seattle, USA (2005)
61. Zhao, W., Liu, X., Li, X., Liu, D., Zhang, S.: Research on hierarchical P2P based RFID code resolution network and its security. In: Proceedings of the Fourth International Conference on Frontier of Computer Science and Technology, Shanghai, China (2009)
62. Zhao, W., Liu, X., Zhang, S., Chen, B., Li, X.: Hierarchical P2P based RFID code resolution network: structure, tools and application. In: Proceedings of International Symposium on Computer Network and Multimedia Technology (CNMT'09), Wuhan, China (2009)