Architecture Evaluation for Distributed Auto-ID Systems

Hong-Hai Do, Jürgen Anke, Gregor Hackenbroich
SAP AG, SAP Research CEC Dresden
Dürerstr. 24, 01309 Dresden, Germany
{hong-hai.do, juergen.anke, gregor.hackenbroich}@sap.com

Abstract
Auto-ID technologies allow capturing the time and location of products in the supply chain for tracking and tracing. This paves the way for a variety of business applications, such as Anti-counterfeiting, Theft prevention, Pedigree, and Genealogy, which analyze the trace history of products to detect relevant patterns or anomalies in the supply chain. While these applications have gained considerable interest recently, further work is needed towards integration of event data from autonomous and possibly heterogeneous Auto-ID nodes in order to obtain the complete trace history for products of interest. As a first step, we perform an architectural study on interoperable Auto-ID systems and present the results in this paper. We first review established techniques for data integration and data sharing as well as relevant industrial efforts. We then clarify the requirements that need to be addressed by an Auto-ID network. Finally, we discuss four possible architecture alternatives for implementing interoperability in such a network and comparatively evaluate the approaches according to the identified requirements.

1 Introduction
In the supply chain, products or materials typically have to travel within and across companies. Each company may utilize different Auto-ID technologies to track and control the flow of the products between the locations, like warehouses, manufacturing and assembling lines, within its boundary. At each location or node, a so-called Auto-ID system (AIS), for example the SAP’s Auto-ID Infrastructure of SAP [9], employs different kinds of sensing devices, e.g. mobile or stationary barcode and RFID readers, to capture Auto-ID information. This information reveals at which point in time a particular product was detected at a particular location in a particular business context, such as the packing or unpacking process for a particular customer. As soon as the objects leave one company and enter another one, their flow may in turn be tracked and controlled by another AIS. The result is a continuous flow of objects, e.g. products or pallets of products, between autonomous AISs as illustrated in Figure 1.

Currently, product information is typically stored at multiple places next to where it has been captured in the supply chain. Simple, yet interesting questions, like “Where has a product been in the supply chain?” or “How many units of its type are still in the supply chain, at which locations?”, cannot be answered, unless the data concerning the product or product class is properly integrated from the corresponding places and analyzed accordingly. Hence, our vision is to establish an interoperable Auto-ID network, the so-called Distributed Auto-ID System (DAIS), which can automatically locate and integrate relevant product data from different nodes to answer such questions. The ultimate goal is the ability to reconstruct complete traces for objects of interest. Detailed product traces are the prerequisite for a variety of business applications, such as Anti-counterfeiting, Theft prevention, Pedigree, Genealogy, Inventory visibility, etc., which aims at detecting patterns or anomalies in product flows.

Figure 1. Object flows captured by autonomous Auto-ID systems
This paper presents an initial architectural study for the envisioned distributed Auto-ID system. We primarily focus on the aspects of data integration and data access, which represent the major functionalities required in order to achieve complete visibility on product traces. In the next section, Section 2, we discuss some related work, including the established approaches for data integration (data warehousing and mediators) and data sharing (P2P), as well as the currently available commercial solutions of EPCGlobal and WWAI, which are still at an early stage. Section 3 elaborates on the technical requirements for the DAIS, especially, issues of data modelling, data integration, and data access in a distributed environment. Section 4 proposes and discusses four possible architecture alternatives for the distributed Auto-ID system. Finally, we conclude in Section 5.
2 Related Work

2.1 Data Integration Approaches

Data integration aims at providing a uniform query interface for a set of heterogeneous data sources. It is needed by the DAIS to integrate event data from different nodes in order to reconstruct complete trace history for objects of interest. This data integration can be achieved by either the materialized or the virtual approach.

The materialized approach physically integrates all relevant data in advance in a central database, the so-called data warehouse [3], which promises significant advantages regarding ease of use but also performance, especially for intensive queries involving large amounts of data. However, the data warehouse needs to be regularly updated with new data. An example for this integration approach is the Business Warehouse of SAP [9].

The virtual approach, also called mediator [11], performs data integration on demand. Given a query, the mediator identifies and queries the relevant sources, and assembles the results returned by the sources to a final result. Up-to-date data is obtained, however, at the price of high effort at query time. An example of this approach is the Virtual Operational Data Store of CenterBoard [1].

2.2 Data Sharing Approaches (P2P)

Peer-To-Peer (P2P) computing [7] aims at a transparent infrastructure for the sharing of computer resources, such as content, storage space, and CPU power. In a P2P network, such resources are contributed by individual peers, which can join and leave the network voluntarily. Peers establish direct interactions with each other to exchange objects or to use them for problem solving. By doing so, they constitute an “overlay” network on top of the physical computer (typically IP) network.

Regarding the centralization of the overlay network, we can differentiate between decentralized and centralized P2P networks. All nodes in a decentralized network (e.g. Gnutella [4]) exhibit exactly the same capabilities and can act both as servers and clients. In contrast, some nodes in a centralized network (e.g., Napster, Edutella [6]) are assigned, either statically or dynamically, a more important role to coordinate activities in the network, e.g., acting as central indexes to facilitate resource lookup.

Orthogonally, we can also distinguish between structured and unstructured P2P networks according to the placement of resources. In an unstructured network (e.g., Gnutella [4]), resources simply stay at their owners, i.e., completely unrelated to the overlay topology. In structured networks, such as Chord [10] and CAN [8], the overlay topology is tightly controlled and resources (or pointers to them) are placed at precisely specified locations, which are typically identified using a hash algorithm mapping resources and nodes to a uniform coordinate space.

The P2P idea is very similar to that of the Auto-ID network, promising the reuse of P2P technologies for the DAIS. However, current P2P networks only support homogeneous resources of coarse granularity, such as files, which can easily be located and shared. Sharing structured data, like product and event data as required in the DAIS, still require further work.

2.3 Current Industrial Solutions

2.3.1 EPCglobal

The EPCglobal effort [2] aims at a standardized interface and architecture for product data exchange systems. It is based on a numbering scheme, the so-called EPCs (Electronic Product Codes), to uniquely identify products in the supply chain. EPCglobal currently works on the specification of a set of network services:

- **Reader Protocol**: The Reader Protocol defines the access to EPC data stored on a product’s RFID chip or captured from sensors.
- **EPC Information Services (EPCIS)**: EPCIS is responsible for storing event data and enabling access to it via web services.
- **Object Naming Service (ONS)**: ONS provides pointers to the EPCIS system of a product’s manufacturer, where static product data can be found.
- **Discovery Service (DS)**: DS is used to determine EPCIS systems of parties, which contain other data, e.g., events, locations, for a given EPC. This can be suppliers, wholesalers and others along the value chain.

The main advantage of the EPCglobal approach consists in the position of a standard and the wide industry support. However, to date, only the specification ONS has been completed in a first version, while the more important EPCIS and DS interfaces are not finished yet. Another issue is that the EPCglobal architecture primarily focuses on exact-match queries based on EPCs.

2.3.2 World Wide Article Information (WWAI)

The World Wide Article Information (WWAI) [12] of Stockway provides an XML-based communication protocol for exchange and querying of product-related data. An implementation of the protocol is currently available, called Trackway. In a WWAI network, there is no centralized storage for any kinds of data or metadata. Each node is an autonomous system, which can grant or prevent access to its data from other nodes.

WWAI follows the structured P2P approach by placing objects of interest, e.g., products, manufacturers, at locations, i.e. nodes, identified by a so-called Dynamic Mapping mechanism. This permits to easily locate nodes potentially storing data for objects of interest. Furthermore, subscriptions can be specified for a particular object ID. Whenever new information on the object becomes available on the node that received the subscription, an automatic notification will be sent to the subscribers.

The main advantages of WWAI include a working implementation of the protocol and the decentralized nature of the solution, allowing for easy implementation and deployment. However, WWAI is a proprietary specification with little industry support. Furthermore, it is unclear how the approach addresses the critical issues of P2P networks, such as ensuring result quality and response time for queries, and limited querying capabilities confined to object ID exact matching.
3 Technical Requirements for the DAIS

3.1 Data Modelling

The first task of the DAIS is to achieve a uniform representation of relevant data for easy integration across different nodes. In general, we can distinguish between product data and event data as explained in the following.

3.1.1 Product data

Product data consists of information that comes with a product and does not change over its life time. It can be further classified into class-level and instance-level data:

- **Class-level data**: This data is common to all instances, e.g., products of a particular product class. Examples include information on the manufacturer, structure, material, etc.
- **Instance-level data**: This data varies from instance to instance within a class. Examples for this data include manufacturing date, lot number, expiration date, etc.

3.1.2 Event data

Event data comprises events captured in the flow of a product in the supply chain. Events can be detected via Auto-ID technologies, such as RFID or barcode readers, or may also be manually entered by human users. The following attributes are typically used to describe the events:

- **Object**: Objects, for which an event is captured, can be a single product or a set of products within a pallet or case.
- **Time**: This attribute captures the time point, at which a product or a pallet is detected at an RFID reader.
- **Location**: This attribute describes the location of the event, typically the location of the reader generating the event.
- **Business process**: This attribute contains the information on the business transaction, in which the event occurs, such as the packing/unpacking process for a particular customer.

While product data is static and remains largely constant over product lifetime, event data is growing over time. Furthermore, product data is known to be heterogeneous and can be modelled in different ways, while event data is mostly homogeneous and easily structured in a tabular format. In the DAIS, event data will be the main focus of analysis, while product data offers additional search criteria to identify products of interest.

3.2 Data Integration and Management

In an Auto-ID network, data concerning the same products is typically scattered over several nodes. In order to reconstruct the complete trace history for such items, data need to be collected from the single nodes and merged under a unified view. This data integration is essentially influenced by the following aspects:

- **Flexibility**: The network of Auto-ID nodes is supposed to constitute a dynamic environment, in which new nodes may be added. The data captured and provided by these new nodes should be easily and seamlessly integrated with that from the existing nodes.
- **Scalability**: An Auto-ID node typically manages a large amount of event data, which is further growing over time. Hence, a major requirement is to provide corresponding optimization mechanisms, such as pre-integration and result caching, in order to achieve reasonable response times for queries across nodes.
- **Security**: Within a company, product traces may be captured at a high level of detail, e.g. at single stations in the assembly line. Such detailed information may be sensitive and irrelevant for other companies. Therefore, it is necessary to address the inter- and intra-company needs for data by providing corresponding views.

3.3 Data Access

The ultimate goal of integrating event data is to support a variety of analysis applications, ranging from control and visualization of product flows to business intelligence on Anti-counterfeiting, Genealogy, etc. These applications in turn pose different requirements on the capabilities concerning data access that need to be supported:

- **Query processing**: It should be easy for the user a) to identify objects of interest, and b) to reconstruct the trace history for these objects. A browsing interface supporting ad-hoc navigation is a useful place for finding further information on the objects. More importantly, a flexible query interface should be supported to allow for search criteria on different attributes of events, such as EPC number, manufacturer, location, time, and business process, i.e. beyond EPC-based exact matching queries.
- **Result quality**: As data is to be collected and integrated from multiple sources for query processing, the major quality criteria for query results include correctness (i.e. the data returned for a query is correct), completeness (i.e., all relevant data is considered in answering the query), and guaranteed response time (in order to avoid the uncertainty about the availability of required data.)
- **Flexible aggregation**: Different applications require different levels of data abstractions. Supply chain control typically requires very coarse visibility on product traces. Other applications, like Anti-counterfeiting, require event history to be documented as detailed as possible. Hence, the flexibility to aggregate event data and to create different views on product traces is needed.
- **Data mining**: In addition to supporting interactive queries of users, which typically have a clear focus on the data to be searched for, data mining should be considered in order to automatically identify patterns or anomalies, which should be examined in detail.

3.4 Comparison DAIS vs. P2P

Although the Auto-ID network may look very similar to a P2P network at first glance, there are still several substantial differences between them, making the application of P2P technologies not always possible or meaningful in the context of the DAIS. We discuss such differences in the following:

- **Cardinality**: An Auto-ID network spanning collaboration within a supply chain is typically much smaller...
than a P2P network, such as for file sharing, which may consists of up to millions of participants (computers). The size of an Auto-ID network may range from hundreds to at most thousands of nodes. For such a small number of nodes, expensive communication protocols, such as, message broadcasting, seems even feasible.

- **Topology:** Although changes may occur in an Auto-ID network over time, it is not expected that its participants, i.e. Auto-ID nodes, enter and leave the network at a fast rate like in P2P networks. This high stability can be exploited in order to reduce the overhead for maintaining connectivity, reorganizing network topology, reallocating content objects or their pointers, etc., as typically required in P2P networks.

- **Granularity:** The objects shared in P2P networks, include files, CPU cycles, or disk storage, are typically coarse-grained, unstructured, and weakly-typed. This makes it easy to identify and share them among participants. In the DAIS, we have to deal with fine-grained structured data, which can be shared at different granularity level (according to the search criteria, which may be defined on multiple attributes). This requires more sophisticated algorithms for identifying and retrieving the data of interest.

- **Quality:** P2P systems focus more on the stability of the network due to the unpredictable behaviour of peers than on the quality of the services (e.g. file sharing). As a consequence, query results may be incorrect and incomplete, and a predefined response time typically cannot be guaranteed. In contrast, the quality of query results, i.e. correctness, completeness, and performance, is crucial in the DAIS, as business decisions may be misled by wrong or incomplete results.

4 Architecture Evaluation for the DAIS

4.1 Architecture Alternatives

Based on the comparative observations above, we distinguish between the architecture alternatives for the DAIS using the following orthogonal criteria:

- **Query processing/Network centralization:** This criterion takes the dependency of Auto-ID nodes on a central instance for query processing into account. In particular, it differentiates decentralized and centralized solutions for the DAIS. In a decentralized solution, each Auto-ID node is fully qualified to answer user queries and responsible by itself for identifying and collecting relevant data from remote nodes, and finally, merging it into a uniform view. In a centralized solution, there are one or several additional servers responsible for query processing and data integration.

- **Data integration/Data placement:** This criterion considers the control of data placement in the network, i.e. the data integration approach employed. As discussed in Section 2.1, we differentiate between materialized and virtual data integration. The former performs a rigorous integration process in advance, i.e. grouping and placing data at a predefined place, for efficient query processing, while the latter retrieves and integrates data on demand for each query.

As shown Table 1, the criteria lead to four architecture alternatives, which are discussed in the following.

4.2 Decentralized Architectures

In these architectures, all Auto-ID nodes possess the same capabilities and behave as a decentralized P2P network: they can act as both server and client. Queries can be asked at each node, which then communicates with other nodes to obtain required data. The solutions can be further classified into unstructured and structured alternatives as discussed below:

#### 4.2.1 Unstructured Decentralized

The unstructured decentralized architecture, also called unstructured P2P, does not control the placement of data in the network. Product and event data are stored at its owner and no reallocation of data is performed until query time, i.e. following the virtual data integration approach. Each Auto-ID node needs to know its neighbours, with which it is directly connected, for message forwarding. The high flexibility of the network to extend and to cope with volatile peers is achieved at the price of a high complexity of query processing.

In particular, to answer a query, an Auto-ID node needs to a) identify all relevant nodes in the network providing required data, and b) request the data from them. Non-deterministic searches, which are typical for this architecture of P2P networks (see [4], [13], [5]), cannot be utilized for the DAIS, as they cannot ensure the correctness, completeness of query results, and a guaranteed response time for queries. Therefore, brute-force search, such as message broadcasting or flooding, is necessary. For a limited number of Auto-ID nodes, this method is still feasible to reach all the nodes and to obtain a complete answer set. However, large amounts of data may need to be forwarded through the network at query time, making it difficult to guarantee a predefined response time of queries.

#### 4.2.2 Structured Decentralized

In the structured decentralized architecture, or structured P2P, all data is also kept at its owner. However, following the materialized data integration approach, newly captured data, i.e. objects or events, is pushed in real or near-real time automatically to other Auto-ID nodes that possibly require the data in future queries. This can be done in two different ways, via either Subscription or Hashing:

- **Subscription:** Each Auto-ID node sends its data requirements, e.g. events concerning products with particular EPC codes or of a particular manufacturer, to other nodes in the network via broadcasting. At the same time, it receives and stores the requirements from other nodes as corresponding subscription profiles. New events can then be sent either immediately or peri-
periodically in bulk to all requesting nodes according to their subscription. Still, if a node needs data that is not yet locally available, it needs to perform brute-force broadcasting of query messages in the network in order to obtain the corresponding data.

- **Hashing:** This approach uses a hash algorithm to evenly distribute data over the nodes in the network, as done in several P2P networks for content sharing like Chord [10], and CAN [8]. Such a hash algorithm identifies for each newly captured event a responsible node, to which the event is to be replicated. However, hashing limits query processing to exact-match queries based on IDs. A remedy for this issue might be to apply the hash algorithm to multiple attributes of events, e.g., object, location, time, and business process, and to propagate events to all nodes identified as responsible for the respective attributes.

### 4.3 Centralized Architectures

Centralized architectures contain one or multiple servers for data integration and query processing purposes. Due to the relatively static nature of the Auto-ID network, it is not necessary that the server role is dynamically assigned to a node, as done in certain types of P2P networks [6]. Each server may be responsible for coordinating a cluster of nodes in a large network. Considering the capabilities of the servers, we further distinguish between solutions, respectively, with a light-weight server, which only focuses on integration at the metadata level, and with a powerful server, which completely takes over the data integration task. The two variants, Metadata Integration Server and Data Integration Server, are discussed in the following.

#### 4.3.1 Metadata Integration Server

Figure 2 illustrates the function of the Metadata Integration Server. In the simplest case, the server maintains only a single index table (EPC, NodeId) storing the association between the EPC numbers and the nodes capturing their events. Similar to the file sharing system Napster, each Auto-ID node maintains a connection to the server and periodically sends a limited amount of metadata on newly captured objects or events to the server, e.g., tuples of (EPC, NodeId) (Step 1). If there are multiple servers in the network responsible for different clusters, they need to update their indices with each other (by periodically forwarding new metadata to the other servers). The update of indices is performed independently from query processing, hence does not affect the performance of the queries.

To retrieve data that is not locally available, an Auto-ID node poses queries to the server responsible for the corresponding cluster. Given a query (Step 2, query of Node 2: \textit{Retrieve events for EPC X in Time Range A-B}), the server queries the index table to identify the nodes (4 and 5) providing event data for EPC X. It then sends the query to these nodes to retrieve the required data, assembles the results (Step 3), and sends the final result to the requesting node (Step 4). The server may also immediately return the list of the relevant nodes looked up in the index to the requesting node, which then queries the nodes and assembles the result by itself.

This approach presupposes powerful query processing capabilities of the Auto-ID nodes in order to answer complex queries involving multiple search criteria. The cost for query processing is optimal if the server is able to identify exactly the nodes that in fact also provide the required data. To improve the accuracy of node identification, the server may maintain additional indices, for example, for business process, time, location, etc., to support search criteria formulated on such attributes. However, there is also an increased amount of effort necessary for updating such indices between the Auto-ID nodes and the servers as well as among the servers in the network.

#### 4.3.2 Data Integration Server

Figure 3 illustrates the principle of the Data Integration Server approach. A Data Integration Server integrates data in advance and stores it locally in order to answer queries without involving other nodes at query time. All Auto-ID nodes maintain a connection to the server and periodically send their newly captured data to the server, i.e., complete records of event data (EPC, Time, Process, …) (Step 1). To obtain data from other nodes, an Auto-ID node poses a query to the server in the cluster (Step 2, query from Node 2: \textit{Retrieve events for EPC X in Time Range A-B}). With the data previously provided by the nodes in the cluster, the server is able to immediately answer the query and send back the result to the requesting node (Step 3).

The propagation of data from Auto-ID nodes to the server is performed independently from query processing, and hence does not affect the performance/response time of queries. Still, it is an expensive operation, as typically large amounts of data are to be sent over the network to the server. Hence, the implementation of a Data Integration Server may only be feasible for clusters with a small number of nodes. Another issue is how to achieve an efficient communication between the servers for processing global queries spanning multiple clusters. A promising approach is to avoid full data replication and only to exchange metadata between the servers for indexing and lookup purposes as discussed for the Metadata Integration Server. That is, a server physically manages all data obtained from the nodes...
to exact-matching queries based on object IDs. Performance generally predictable. However, it is restricted to ensure a good load balancing across nodes making query equal to that in the unstructured P2P approach. Hashing can be supported by an index (EPC number, Node ID). Query performance depends on the coverage of the indices: if indices on different kinds of information are available, relevant nodes can be identified accurately according to the search criteria of queries.

The Data Integration Server approach requires the highest effort for data integration. However, due to the local availability of data, a Data Integration Server promises a number of advantages that are not possible with the other approaches. These advantages include ability to process complex queries beyond object ID-based exact matching, high query performance, flexibility to provide different views/aggregations of data for different analysis tasks and security reasons, local availability of data for data mining, and central control over data archiving and purging.

5 Conclusions

The paper presented the first results of an architectural study for distributed Auto-ID systems, which aim at integrating event data from autonomous Auto-ID nodes to obtain trace history of objects in the supply chain. We surveyed related technologies for data integration and P2P as well as industrial efforts, while the latter are still at early stage. We identified and discussed the major requirements for the DAIS. We then outlined different architecture alternatives for the DAIS and used the identified requirements for a conceptual comparison. Each alternative has its own advantages and disadvantages. In future work, we intend to define a set of test cases for practically assessing the feasibility and performance of each architecture approach.

Acknowledgement. We thank Jochen Rode for helpful comments on an early draft of this paper.

References