Discovery Services in the EPC Network

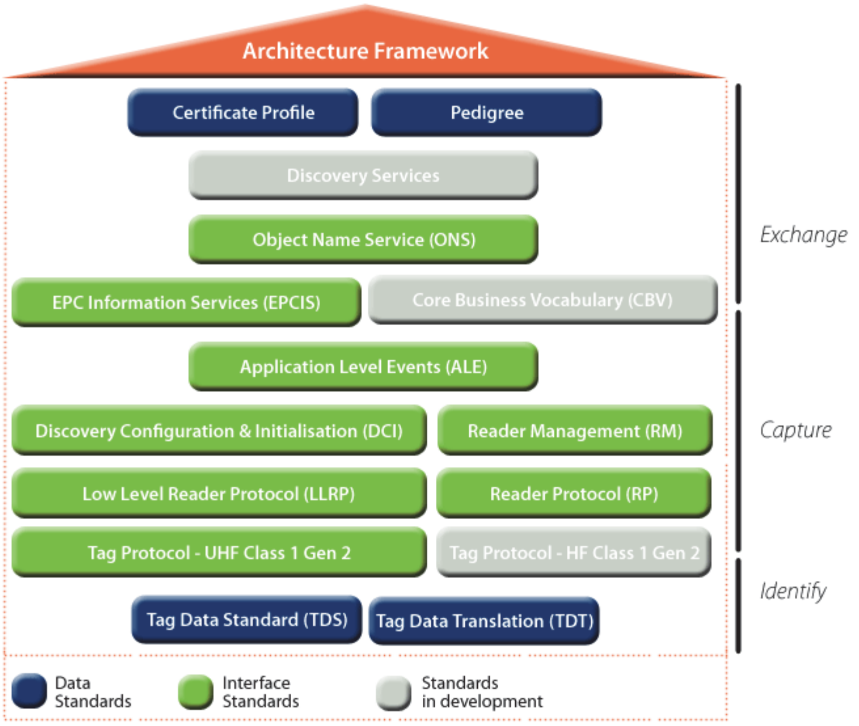
Martin Lorenz, Jürgen Müller, Matthieu-P. Schapranow,
Alexander Zeier and Hasso Plattner
Hasso-Plattner-Institute
Germany

1. Introduction

Recent advances in Auto-ID technology, especially RFID, provide great potential for the innovation of existing processes in Supply Chain Management (SCM). Accompanied with item level identification using the EPC, companies are able to capture product lifecycle information at unprecedented levels of detail. RFID readers placed at strategic points in the supply chain automatically capture information about passing objects while they move along their way from the manufacturer to the consumer. Modern RFID tags can be equipped with sensors for temperature, humidity or other physical conditions, providing information systems with instant data on the current location and status of objects. Auto-ID bridges the gap between the physical and the digital world, providing real-time information about current supply chain operations. It provides companies with increased supply chain visibility [Melski et al. (2008)], resulting in reduced uncertainty, regarding operational and tactical supply chain planning. Overall, Auto-ID supports companies by providing higher information quality and quantity.

While most of the aforementioned aspects concern company internal processes, an even greater potential is being anticipated for company-overlapping supply chain collaboration. The possibility to provide real-time information about intra-company operations to trading partners, up- and downstream the supply chain, allows companies to increase value creation over all levels of the supply chain. In particular, planning activities of adjacent trading partners can be performed with a higher degree of certainty, reducing the need for high safety stock levels, which in turn reduces inventory costs [Simchi-Levi et al. (2003)]. On the other hand, many industries struggle with volatile demands, leading to the risk of running out of stock in times of higher demand. Real-time information can help to detect critical stock levels early. Sharing that information instantly with suppliers allows them to take immediate action such as rescheduling of shipments or increasing production rates to cope with temporary increased demand. Section 2 of this chapter will go into the details of two selected industry use cases that outline the benefits of company-overlapping collaboration.

The existence of practical scenarios for supply chain collaboration based on Auto-ID data demands for an infrastructure of information systems to support these use cases. EPCglobal, a joint venture between GS1 (formerly known as EAN International) and GS1 US (formerly the Uniform Code Council, Inc.), introduced the EPCglobal Architecture Framework, which is suppose to increase visibility and efficiency throughout the supply chain as well as to guarantee higher quality information flow between companies and their trading partners [EPCglobal (2007a)]. The EPCglobal Architecture Framework, for the rest of this chapter named EPC Network, is derived from the concept of the "Internet of Things" (IoT). The IoT is a concept that describes a self-configuring wireless network of sensors whose purpose is to provide objects with a means to interconnect and to interact [Polytarchos et al. (2010)]. Based on this idea, the EPC Network defines information systems, communication protocols, and data types that support capturing, storage, and exchange of EPC data among participants of a supply chain network. Figure 1 depicts the different standards defined for the EPC Network. The architecture includes specification for low level communication protocols such as the air interface between tag and reader as well as high level aggregated business information such as the EPC Information Services (EPCIS) and the EPC Discovery Service (EPCDS). Especially the latter play key roles for the company-overlapping exchange of information.



Architecture Framework
Certificate Profile
Pedigree
Discovery Services
Object Name Service (ONS)
EPC Information Services (EPCIS)
Core Business Vocabulary (CBV)
Application Level Events (ALE)
Discovery Configuration & Initialisation (DCI)
Reader Management (RM)
Low Level Reader Protocol (LLRP)
Reader Protocol (RP)
Tag Protocol - UHF Class 1 Gen 2
Tag Protocol - HF Class 1 Gen 2
Tag Data Standard (TDS)
Tag Data Translation (TDT)
Exchange
Capture
Identify
Data Standards
Interface Standards
Standards in development

Fig. 1. EPCglobal Architecture Framework

The diagram depicted in Figure 1 shows the discovery service component in a pale green color, indicating that it is still question to research how such a discovery service has to be designed. The purpose of this chapter is to elaborate on the complexity of this issue and introduce scientific work related to the definition of a discovery service component for the EPC Network. There are numerous functional and non-functional requirements that make the definition of an application layer protocol for a discovery service a difficult task. In Section 2, we present real world use cases that require the existence of a discovery service, to substantiate the necessity for such a component. In Section 3, we take a closer at the EPC Network components that are needed to support the use cases described in 2. Subsequently, we enumerate requirements for a discovery service to support the presented use cases. Based on these requirements, we propose a discovery service design for the EPC Network in Section 5. Section 6 gives an outlook on future work.

2. Industry use cases

To stress the need of a discovery service for the EPC Network, we present two real world industry use cases in this section. We do this for two reasons. First of all, practical use cases proof the necessity of a research topic, regarding its significance to economic interest for industries. Secondly, use cases can be used to derive concrete requirements for the design and the implementation of an information system. For this purpose, we introduce an anti-counterfeiting scenario in the context of the European pharmaceutical supply chain in Section 2.1, and we describe the process of product recalls in Section 2.2, focusing on the localization of effected products to provide effective recall management, keeping the financial impact as low as possible.

2.1 Use case 1: Anti-counterfeiting

As production in low-wage regions and global trade increases, opportunities for producing and selling counterfeit products also arises. The Organization for Economic Co-operation and Development (OECD) conducted a comprehensive study in 2008 [OECD (2008)], which was updated in 2009 related to the economic impact of counterfeiting and privacy [OECD (2009)]. It estimates that the trade volume of pirated and counterfeit goods could sum up to $250 billion excluding domestically produced and consumed products and pirated digital products. This is an equivalent of 1.95% of the world trade volume.

This poses a financial risk to companies because fake or smuggled goods reduce their sales volume. The pharmaceutical industry moved to public focus by the operation MEDI-FAKE, conducted by custom authorities in all EU members states. More than 34 million fake drug tablets were detected at customs control at the borders of the European Union in a two month period [Group (2009)]. This can put lives in danger as pharmaceuticals might not contain active pharmaceutical ingredients, wrong ingredients, a wrong dosis or other harmful substances.

To increase process efficiency and fight smuggling as well as counterfeiting, companies more and more inspect the concept of "unique identification", meaning that not only the product manufacturer and the product type is encoded but that each and every single item receives a unique serial number. That is the point where EPC an RFID comes into play. With the ability of unique identification using EPC and ubiquitous data capturing using RFID, it is possible to track items along their way from the point of production to the consumption. A major component in such a scenario is the company's read event repository, which stores the events captured by the RFID readers. Each company in the supply chain that captures Auto-ID data from their processes, needs to operate such a read event repository, to persist its data. Combining the information distributed over all repositories of the companies that are part of the manufacturing and/or distribution process, allows to reconstruct a complete trace of each individual item. Such a trace can be used to verify the origin and the distribution path of an item, providing customers only with pharmaceuticals from licit supply chains.

The problem is that a retailer needs to determine all resources of information, i.e., the addresses of the read event repositories that contain information regarding the particular EPC. Globalized trade, dynamic business relations, re-importing, and multiple levels of wholesalers and distributors, require a dynamic aggregation of information from a number of potentially unknown resources. To gather all this information, a component is needed that, given an EPC, provides pointers to the resources that contain the read events created during the travel of the item through the supply chain. Such a component is the EPC Discovery Service.

2.2 Use case 2: Product recall

The second use case that we want to present is product recalls. Product recalls usually occur due to safety or quality issues. They require a higher planning effort than most other return types. Key to a successful management of recalls is information technology and effective communication. Product recalls can be voluntary or mandated by legal obligations. A recent example is Toyota's production problems in October 2010 [Ohnsman & Kitamura (2010)]. They had to recall 10 million vehicles globally, because particular models might have brake system and gas pump issues. For many industries that are susceptible to recalls, like the automotive or food industries, a poorly managed recall can create a tremendous negative impact on the economic side of the company. Even more problematic is the accompanying damage in reputation, which can become a threat to existence.

In such a scenario like in the case of Toyota, it is most important to determine the exact number of affected products to act fast and target-oriented to contain the potential financial damage. In most cases not all of a company's products need to be returned. Temporary production problems in one of the production plants might have caused a subset of all products to be erroneous. Consequently, the company needs to find out where these products have been and who they have been sold to. That way it is possible to keep the number of recall products as small as possible, recalling only the ones that have been identified as potential defects.

Using RFID and EPC, it is possible to trace the distribution of each individual product. In case of food or life stock, it is also possible to determine all products that the item has been in contact with during storage or transportation, eliminating the possibility of collateral damage due to dispersion of poison or illness.

Again, this information is distributed over a number of independent read event repositories, which are operated by the companies that traded the goods. To perform effective product recall, we need to aggregate and analyze all the information distributed among the resources. Just like for the anti-counterfeiting scenario, a discovery service needs to be present to enable such kind of innovative process.

Now that we presented industry scenarios where Auto-ID technologies can help a great deal to improve current processes, we want to take a closer look at the EPC Network and the components that are needed to support our ideas.

3. EPCglobal architecture framework components

The previous section described practical use cases for a discovery service for the EPC Network. In this section, we go into the details of the EPC Network to understand the interconnection between the individual components and their relation to the use cases. We need to do this because most of the requirements for a discovery service are based on the existing components, the data that is available in the network, and the interfaces used to access the data. We will not go into the details of low-level physical data access and tag encodings, instead we restrict our discussion to the components above Application Level Events (ALEs), see Figure 1.

3.1 Read events

The primary type of data exchanged in the EPC Network are read events. read events are business-level events, which represent a scan of an RFID tag or 2D barcode associated with business context. There are five types of events: EPCISEvent, ObjectEvent, AggregationEvent, TransactionEvent and QuantityEvent. Figure 2 depicts an UML class diagram, showing the relation between the different types of events.

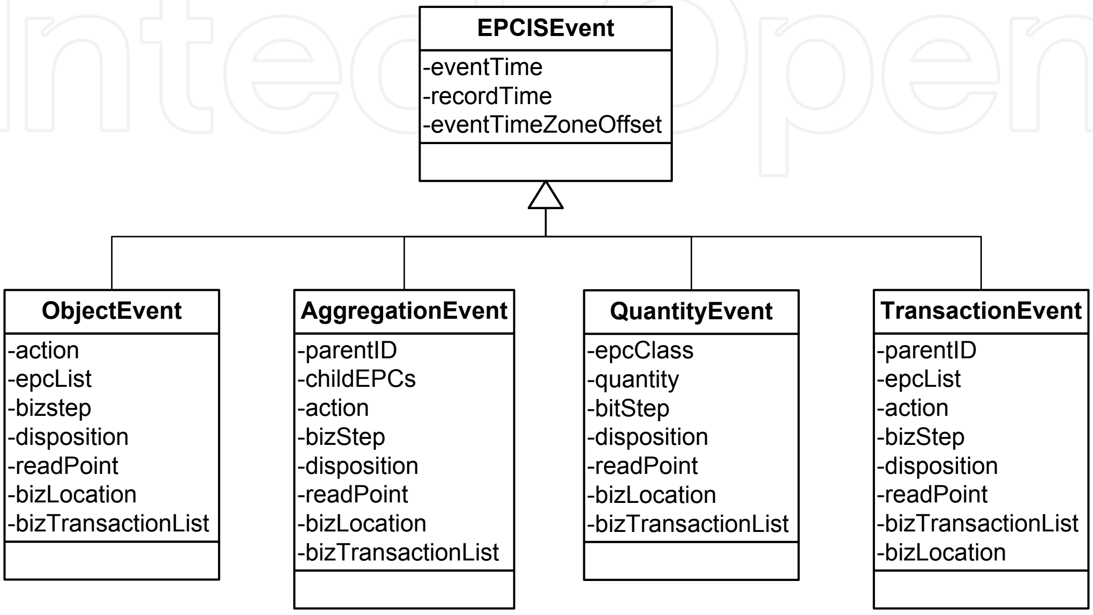


Fig. 2. Class Diagram of EPC Event Types

These events answer the questions *What*, *Where*, *When*, and *Why*. The EPCglobal standard allows to extend these data into each direction to provide companies with the ability to adapt the data to their special needs. For a detailed discussion on the meaning of the individual attributes of the events, we point the interested reader to the EPCglobal EPCIS standard [EPCglobal (2007b) (Section 7)]. With these read events, it is possible to identify location and business context of items during their travel through the supply chain.

3.2 EPC information services

Once these events are created, they need to be stored persistently at some point, to provide other applications with the ability to use these events. For this purpose, the EPC Network defines the EPC Information Services. The EPCIS provides a repository to store the information about read events that is why it is also called read event repository. Furthermore, it provides a capture interface to provide a way to store the events, as well as a query interface to query for stored events. Each company, which captures Auto-ID data is supposed to operate an EPCIS to be able to store and to exchange the information with internal and external applications. Figure 3 illustrates the process of information storage and exchange with the EPCIS. However, the EPCIS is nothing more than a repository for read event data. It solely serves as a resource of information and does not implement any business logic. In order to be able to leverage the full potential of the information distributed among the EPCIS servers of different trading parties, it is necessary to derive the exact addresses of the EPCIS servers that posses information about a particular item, i.e., EPC. The EPC Network defines two

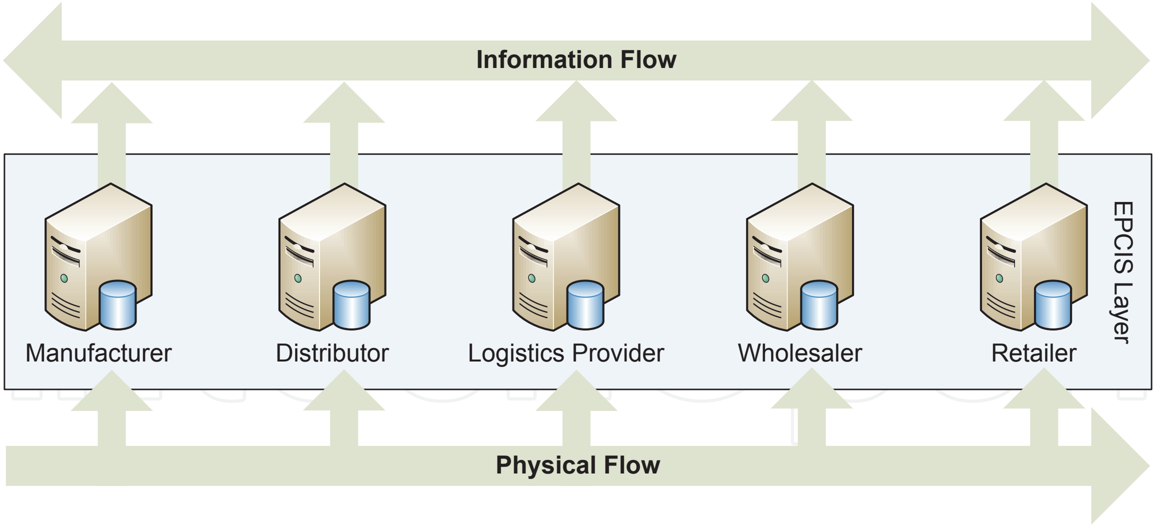


Fig. 3. EPC Information Services Data Flow

information systems that provide such kind of functionality, namely the Object Name Service (ONS) and the EPC Discovery Service.

3.3 Object name service

The ONS is a DNS-based service, whose purpose is to resolve information resources to an EPC. Information resources in the context of ONS can be websites, web services, or an EPCIS repository. However it is important to note that the ONS does not process the serial version of the EPC. Figure 4 depicts the EPC numbering scheme. It consists of a header, defining the version of the EPC, an EPC manager, identifying the authority that assigned the EPC to the object, an object class, which identifies the type of object, and a serial number, used to identify a particular item among a number of items of the same class and manager. The ONS neglects the serial number of an EPC [(EPCglobal, 2008, Section 5.2.1)]. The granularity of ONS resolution is currently limited to product type, rather than serial-level lookup. i.e. an ONS is not expected to retain distinct records for two objects of the same product type that only differ in their serial numbers. The only EPCIS server address that is being stored by the ONS is the manufacturers EPCIS, where the EPC has been assigned to the item. So if we want to store a list of different EPCIS server addresses for an individual item, we need another information system.

01.0000A89.00016F.000169DC0

Header EPC-Manager Object Class Serial Number

Fig. 4. Structure of the EPC Numbering Scheme

3.4 EPC discovery service

The EPC Discovery Service standard is currently in development by the EPCglobal Data Discovery Working Group. Its main purpose is "Finding and obtaining all of an item's relevant visibility data, of which a party is authorized, when some of that data is under the control of other parties with whom no prior business relationship exists" [EPCglobal (n.d.)]. The EPCDS can be seen as a search engine for EPC-related information. Given an EPC, it returns a list of URLs of the query interfaces of EPCIS servers, which are in possession of information related to the particular EPC. With this functionality, authorized and authenticated clients are able to reconstruct traces of items and to track the current location of items. Figure 5 illustrates the semantic difference between ONS and EPCDS. Looking at the use cases from Section 2, only the EPCDS provides enough functionality.

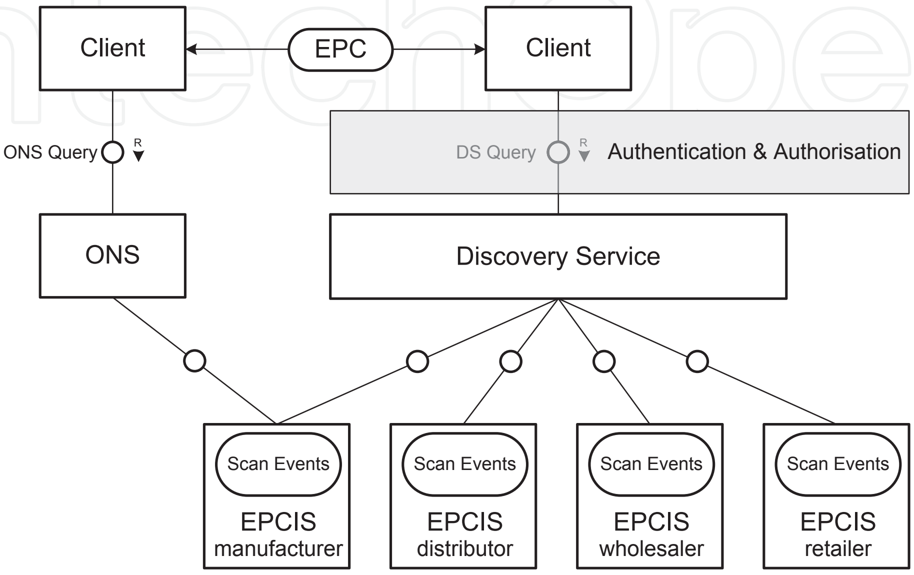


Fig. 5. Object Name Service vs. EPC Discovery Service

In this section, we looked at the individual EPC Network components and defined their particular roles, regarding information storage and exchange. We introduced the concept of the EPC Discovery Service, which is the central component to support the use cases from Section 2. The following section takes the prerequisites from this section and the use case definitions and derives a list of basic requirements for a discovery service for the EPC Network.

4. Discovery service requirements

In order to create an architecture design proposal for a discovery service, we need to define a set of requirements. This section enumerates requirements, which are used in Section 5 to reason on the design of our proposed discovery service architecture. The requirements have been gathered and consolidated from a number of resources. First and foremost, we used the results of the BRIDGE project, which is an integrated Project addressing ways to resolve the barriers to the implementation of the EPCglobal Network in Europe. Work package two of this project accessed requirements and designs for a serial-level lookup service for the EPC Network. Furthermore, we have taken argumentations from Müller et. al. [Müller, Oberst, Wehrmeyer, Witt & Zeier… (2009)] and Kürschner et. al. [Kürschner, Condea & Kasten… (2008)], which contribute to create a comprehensive set of discovery service requirements. These requirements include the main topics core functionality, data ownership, security, business relationship independent design, organic growth, scalability, quality of service, client complexity, and bootstrapping.

4.1 Core functionality

At its core, a discovery service needs to store the EPC, which has been observed, the URL of the EPCIS server that stores the actual event and a timestamp. In order to store this data, the EPCDS needs to offer a notification interface that can be used by resources to publish their information. Additionally, there needs to be a query interface, which allows clients of the discovery service to request the stored information. Parallel to this query interface, which allows ad hoc queries, there should be a way to register standing queries, which provide companies with the ability to get instant information on incoming notifications. Since the information, stored at the discovery service, is highly sensitive to companies, there should also be a security component in place that implements authentication and authorization functionality, to protect the data. The following enumeration summarizes the core functional requirements of the EPCDS labeled from **RQ1** to **RQ5**.

RQ1: A discovery service needs to provide a way for resources to publish their information, i.e., EPC and corresponding EPCIS server address.

RQ2: It needs to store the EPC/URL mappings and the according timestamps persistently.

RQ3: It needs to provide a way for clients to execute ad hoc queries for EPC-related information.

RQ4: It needs to provide a way for clients to register/unregister standing queries to provide instant information on incoming notifications.

RQ5: It needs to provide authentication and authorization mechanisms to protect the stored data.

4.2 Data ownership

According to Kürschner et al., data control aspects have to be considered by any discovery service approach. Their investigations showed that there exist companies that are not willing to share their EPCs or EPCIS addresses with other companies. The reason for this is self-interest, i.e., system owners have greater interest in system success than non-owners. The issue of data ownership is considered to be a major reason for managers to decline the participation in supply chain overlapping business collaboration. Neglecting this fact will lead to a reduced adoption rate of the particular discovery service approach among supply chain partners. Based on their findings, Kürschner et al. defined two requirements for the discovery service design, regarding data ownership.

RQ6: Companies shall be in complete control over their data including EPCIS addresses, read events, business data as well as setting of detailed, fine-grained access rights.

RQ7: Companies shall be able to track the usage or the requests upon their data. Particularly, publications of data at the discovery service level should be avoided.

4.3 Security

Security is a vital factor in any enterprise application. In case of the discovery service this issue becomes even more relevant due to the fact that it operates on public networks, keeping sensitive information potentially necessary for business success. Kürschner et al. derive a set of characteristics from the overall topic of security. These are availability, reliability, safety, confidentiality, integrity, and maintainability. Although all of the above mentioned characteristics are essential features of a discovery service, only three of them are regarded as outstanding for the design of a discovery service.

RQ8: The confidentiality of both the publisher data and client query shall be ensured by the discovery service design.

RQ9: The discovery service architecture shall ensure a high overall system availability and reliability.

Additionally to the security requirements **RQ8** and **RQ9**, we consider data integrity as a fourth characteristic. Business relations on the level of supply chain management rely on trust. In a collaborative effort to increase the efficiency of modern supply chains, managers base their decisions on data delivered by supply chain participants that have been categorized as trusted partners. To keep these trust relations valid over digital cooperation, there need to be mechanisms to verify the correctness of origin and integrity of the data.

RQ10: The discovery service design shall ensure the correctness of origin and the integrity of the shared data.

4.4 Business relationship independent design

Different customer demands, globalization, discovery of uncharted market opportunities, outsourcing, innovation and competition are some of the major factors that determine significant partner changes in supply chains. From a strategic point of view, for some companies, changing supply chain partners is simpler and cheaper than changing internal processes. Section 4.2 adduced the need for information ownership and fine-grained access control for information sharing. Companies that modify their trading partners relations frequently need to define access rights, reflecting these new business relationships. Having this in mind, it is important to minimize the access control maintenance effort for companies.

RQ11: Changes in business relationships shall not affect the way in which a company interacts with the discovery service.

4.5 Organic growth

Organic growth as used by Kürschner et al. is derived from a definition by Rogers in [Rogers (1995)], where he categorizes adopters of new ideas into innovators (2.5%), early adopters (13.5%), early majority (34%), late majority (34%), and laggards (16%). As a result of this development there will be only few companies initially joining the network in the beginning. However, the actual value of the EPC Network depends on the number of participating companies. Consequently, it is of high importance to lower the threshold for joining the network for less innovative companies, fostering the adoption of the EPC Network.

RQ12: The discovery service architecture shall encourage participation in the EPC Network.

Although this requirement is somewhat straightforward for any new technology, it is worth special consideration, because the value of the network and therefore the acceptance of the EPCglobal idea, to support supply chain innovation for all industries, depends on the fast adoption of discovery services.

Low threshold in this context can be related to technical, financial and political obstacles. In order to push the desire to participate in an innovative idea such as the EPC Network, it is important to keep a positive relation between opportunity and risk. An economically expensive solution, creating large administrative overhead, leads to a low adoption rate, resulting in an EPC Network with low attraction to potentially interested parties.

4.6 Scalability

Another very important requirement is scalability. Müller et al. have already been aware of the problem of handling large amounts of requests and data. The issue of information production in RFID-enabled supply chains has been topic to a number of research works all aiming to understand the nature and behavior of these RFID enabled supply chains [Ilic, A. Groessbauer and & Fleisch (2009)]. Depending on the industry and application scenario, a discovery service can become a bottleneck or, even worse, a single point of failure when scalability becomes an issue.

RQ13: The discovery service architecture shall be highly scalable to be able to handle both, data volume and number of participants.

4.7 Quality of service

From a client's perspective, quality of service means the discovery service needs to provide data that is accurate, complete, and delivered within acceptable time frames. In this context, the predicate ″accurate“ means that the response of the discovery service contains all information, necessary to perform the desired queries on the individual EPCIS servers. Response time is also an important characteristic. Research showed that the acceptable time for an ad hoc query is only a few seconds [of Cambridge et al. (2007)]. Completeness of the result means that the discovery service's response contains all information available in the network and accessible to the client with regard to access control rights. From these findings, requirement number fourteen is derived.

RQ14: The query result shall be complete and correct, respecting the clients' access rights defined separately by each information provider.

Since the original requirement does not contain the dimension time, we propose a second version of this requirement.

RQ14a: The query result shall be complete, correct and within an acceptable time frame, respecting the clients' access rights, defined separately by each information provider.

4.8 Low client complexity

Client complexity is an important requirement because it has a great impact on the usability of the discovery service. It directly determines the interaction behavior between client and discovery service, potentially aggravating possible use cases. This can lead to a low adoption rate.

RQ15: Client complexity for discovery services shall be as low as possible, without loosing functionality.

4.9 Bootstrapping

One of the major concerns for the implementation of successful discovery services is the bootstrapping process. The bootstrapping process enables an interested and authorized client to locate an object's discovery service, using only the object identifier, i.e., the EPC. For many reasons, this is a serious problem. First of all, the plain amount of data produced by RFID-enabled supply chains and the number of queries, requires to operate a number of distributed discovery services, to share the work load [Ilic, Groessbauer, Michahelles & Fleisch (2009); Müller, Pöpke, Urbat, Zeier & Plattner (2009)]. Secondly, there are political problems that prevent the successful operation of a single global discovery service. Companies from many different countries and industries would have to agree on publishing their data to a discovery service, operated by some authority organization. It is most likely that there are countries and individual organizations that are not willing to publish their data to such a discovery service for a number of political reasons. Thirdly, the operation of a global discovery service would require processing power and storage space similar to the data processing centers of the major search engine providers. However, search engine providers are able to be financed via advertisements and additional services. An organization running a global discovery service would have to be financed by its users, who might not be willing or able to pay for the service. This issue directly influences requirement seven.

In the above paragraph, we identified technical, political and economical problems that lead to a distributed network of independent, collaborating discovery services. These discovery services will be operated by different providers such as legal authorities, companies themselves, or third-party profit organizations. In [A. Rezafard (2008)] Rezafard assumes that there will be globally operating communities (supply chains) that commit to a discovery service of choice.

It has been suggested that the ONS could be used for the bootstrapping process. However, the ONS is authoritative in that the entity that has change control over the information about the EPC is the same entity that assigned the EPC to the item to begin with. This means that the entity that assigned the EPC has to determine the discovery service that each company, which gets in contact with the object, has to publish its information to. This procedure may be feasible for supply chains, completely owned by a single company, but it is not possible to force all supply chain participant in global dynamic supply chains to publish their information to a particular discovery service. We already mentioned the issue of information ownership. Each company, producing RFID data is in full control of the data and decides autonomously about the publication of this information. That way it might be possible that the information about an EPC is distributed over a number of different discovery services.

Until now there is no accepted network architecture for discovery services. The reason for this is the fact that there is no common understanding about the distribution of EPCs among discovery services as introduced above. An industry-wide agreed distribution schema for EPCs is the basis for the design of a network architecture for discovery services. Once there is an agreed network structure, it is possible to develop bootstrapping mechanisms that enable supply chain partners to determine suitable discovery services just by means of the given object identifier, e.g., EPC. Recent research proposes Peer-to-Peer overlay networks as a promising way for discovery services to collaborate [A. Rezafard (2008); Shrestha et al. (2010)].

RQ16: The discovery service architecture should support communication of independent discovery services, serving distinctive concerns of individual companies.

RQ17: The network of discovery services should provide a bootstrapping strategy for clients to approach the correct discovery service, only by having the EPC at hand.

5. Discovery service architecture design

In this section we summarize and evaluate existing theoretical and practical discovery service approaches. We reason on their suitability for the EPC Network. Afterwards, we present a discovery service architecture that we designed and implemented prototypically, followed by a comparison of the different approaches with our new design.

5.1 Existing discovery service approaches

Here, we present research related to the definition of a discovery service design for the EPC Network. The different theoretical and practical approaches described below, contribute to our own approach, presented in 5.2.

5.1.1 Beier et al.: Discovery services

In [Beier et al. (2006)] a first implementation of a discovery service is presented. It can be summarized as Directory Look-up approach. In their paper, Beier et al. analyze the appropriateness of the Object Name Service [EPCglobal (2008)] and come to the result that this approach is improper for building discovery services. The developed Directory Look-up approach works as follows: real-world items attached with an EPC travel through the supply chain. At each company the item passes, the EPC is read and a read event is stored in the company's EPCIS. For each EPC that is stored in an EPCIS for the first time, the discovery service is notified and stores the EPC, the URL of the submitting EPCIS, a timestamp, the certificate of the submitter, and a visibility flag in its repository. The discovery service can then be queried with an EPC of interest. It replies with a list of relevant EPCIS URLs. Finally, the requester can query all relevant EPCIS servers by himself and aggregate the respective information. The underlying assumptions are that all participants of the EPC network are authorized by EPCglobal and equipped with a certificate by a trusted third party.

According to Beier et al. [Beier et al. (2006)], access to a company's EPCIS should be implemented role-based and policy-based with cell-level data disclosure control. At the discovery service level, row-level data access control should be enforced and, using the visibility flag, the owner of the data decides whether the record is shared among all authorized participants of the EPC network or access is restricted to companies, which have information about the same EPC. To retrieve EPCIS addresses confidentially, Beier et al. propose the usage of EPCIS proxy servers by storing not the real but the proxies URL at discovery service level.

5.1.2 BRIDGE project: High-level design for discovery services

BRIDGE is an acronym for Building Radio frequency IDentification for the Global Environment. The objective of this EU-funded project is to "research, develop and implement tools to enable the deployment of EPCglobal applications in Europe" [of Cambridge & UK (2007)].

In the report [of Cambridge & Research (2007)] the authors propose eight discovery service approaches, evaluate them, and finally judge four as promising candidates for large scale discovery services. It is important to understand that EPCIS servers can serve two different types of queries: ad hoc queries and standing queries. One-off queries are performed by a client once and no further communication between client and EPCIS is planned. Standing queries are subscriptions, which can be time-controlled using a query schedule (e.g., a client

wants to be informed every hour) or trigger-controlled (e.g., a client wants to be informed if new information about an EPC of interest is available) [EPCglobal (2007b)].0
We will now briefly describe the four candidates identified by the authors. The first candidate is called Directory-of-Resources and equals the Directory Look-up approach by Beier et al. [Beier et al. (2006)]. The second candidate is called Notification-of-Resources and works as the Directory-of-Resources except that a client shows interest about certain information by creating a subscription at the discovery service. Once an EPCIS notifies the discovery service about an EPC, which matches the criteria defined in the subscription, the discovery service informs the client that the respective EPCIS is in possession of information related to the subscription. The third candidate is called Notification- of-Clients: EPCIS servers notify the discovery service for each new EPC they own information about. Once a client shows interest in an EPC the Discovery Server informs all relevant EPCIS servers. The servers send an availability notification to the client, which then queries the respective EPCIS servers. The last candidate identified by the authors is called Query Propagation and acts like Notification-of-Clients except the information is sent to the client by the EPCIS servers immediately without the availability notification. The authors summarize the first two candidates as Directory Service approach and the last two as Query Relay approach.

5.1.3 Kürschner et al.: Discovery service design in the EPCglobal network

In their related work, Kürschner et al. describe that the concepts of the Domain Name Service and the Service Location Protocol are not appropriate to solve the discovery service problem [Kürschner, Condea & Kasten… (2008)]. The authors present the Directory Look-up approach by Beier et al. and criticize that the EPCIS address of companies having information about the EPC of interest might be revealed if there are no access control policies in place. Otherwise, if these policies were established, the maintenance effort and complexity would rise because fine-grained access rights would have to be defined at discovery service level and policies would have to be synchronized between companies' EPCIS servers and the discovery service. To fulfill the requirement of low access right maintenance effort, they present the Query Relay approach developed in [of Cambridge & Research (2007)] in detail. The idea is to use the discovery service as a relay by forwarding the respective client queries to all relevant information holders. The EPCIS servers reply directly to the requester. Therefore, the EPCIS address is not revealed to the requester if the respective company decides not to reply to the query at all. Finally, both presented approaches are discussed and evaluated.

5.2 Our new design - an aggregating discovery service

The idea of the *Aggregating Discovery Service (ADS)* is to forward client queries to relevant EPCIS servers, aggregate their responses and synchronously respond to the client request. This reduces client complexity, brings low response latency, delivers complete and correct information for the requester, ensures data ownership for the information holder, avoids the need for fine-grained access control replicated at discovery service level, and guarantees confidentiality of clients and information holders. The ADS is a centralized service, which offers two interfaces (see Figure 6).
The *query interface* is used to gather information about an EPC of interest from the EPC Network. The ADS links EPCs to supply chain partners, which can provide detailed information about those EPCs. Certificates are used to provide authentication as proposed in [Beier et al. (2006); Kürschner, Condea, Kasten & Thiesse (2008)].

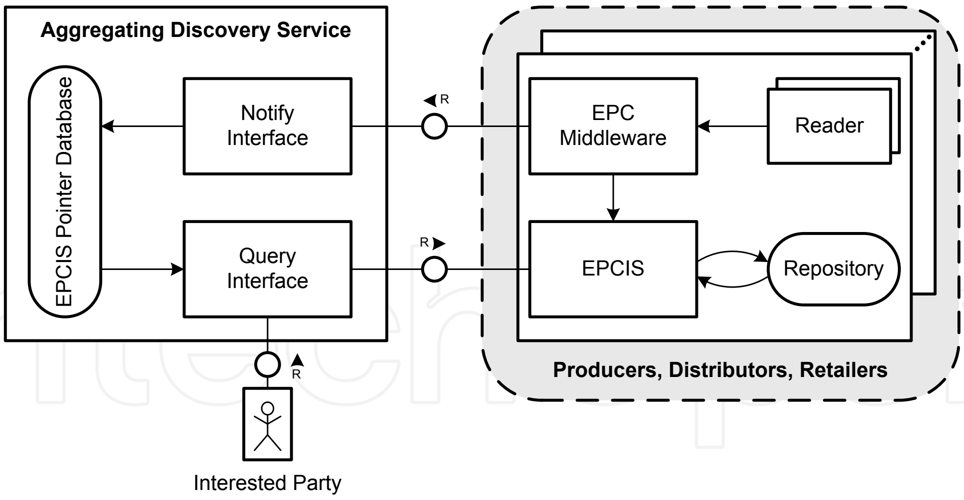


Fig. 6. Aggregating Discovery Service Architecture

The *notify interface* is used to inform the ADS about read events to be shared within the EPC Network. The ADS receives the EPCIS URL of the submitting partner and one or more EPCs that have been handled by this entity. Submitting multiple EPCs at once allows the client to batch notify requests and improves performance by lowering connection overhead. We propose a simple, XML-based format for this message to be submitted via HTTP POST.
The ADS maintains an association between submitting EPCIS servers and submitted EPCs. This allows the ADS to determine all EPCIS servers that hold more information about an EPC. The query relay provides an EPCIS-equivalent query interface [EPCglobal (2007b)] as proposed in [Kürschner, Condea, Kasten & Thiesse (2008)]. Additionally to a *full query* the client also can identify relevant EPCIS servers using a *resource query* [of Cambridge & Research (2007)]. For both types of queries the execution is as depicted in Figure 7.

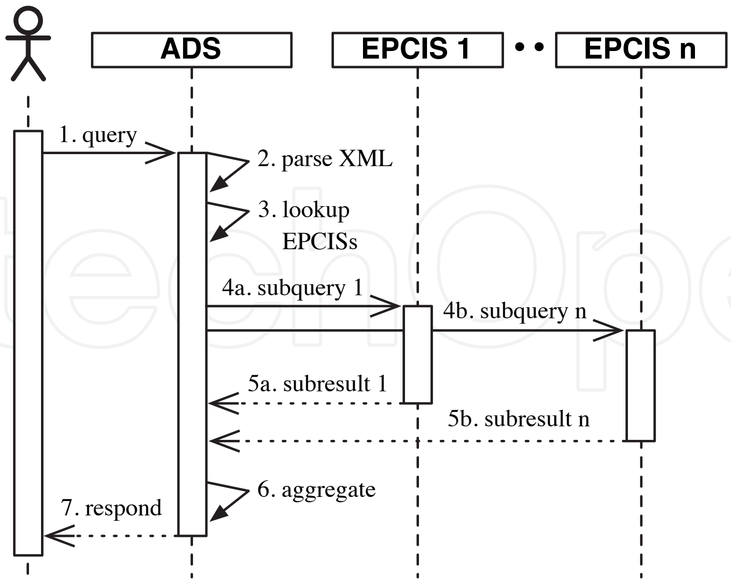


Fig. 7. Client Query Execution

The ADS waits for an incoming client query (*1.*) and parses the query to extract relevant EPCs (*2.*). The ADS then queries its internal database to look up the URLs of EPCIS servers, which are relevant for this query (*3.*) and forwards the original query to those EPCIS servers (*4.*). After subresponses returned from the EPCIS servers (*5.*), they are parsed and the read events are extracted and combined (*6.*). The aggregated result is then returned to the client (*7.*). Effectively, this means the ADS acts as a proxy.

When querying EPCIS servers, problems might occur. Subqueries might time out, EPCIS servers may be temporarily unreachable or may refuse to answer the query. To prevent timeout of its client connection, the ADS will return a possibly incomplete result set marked as such, distinguishing between temporary problems (indicating that the client should try again later) and permanent reasons that prevent returning a complete response.

The ADS query interface should also support standing queries. The ADS needs to store all standing queries it received in order to forward them to an EPCIS when the EPCIS sends an event notification containing an EPC that matches a standing query of a client. This approach has the advantage that other EPCIS servers are not burdened with irrelevant standing queries, as they would be if the ADS was to distribute all standing queries to all EPCIS servers in order to achieve complete coverage.

5.3 Comparison of the different approaches

The fulfillment of the requirements stated in Section 4 and referenced literature in that section is substantial for a well-designed discovery service architecture that can easily be integrated in the EPC Network. In this section, we compare the existing approaches, introduced in Section 5.1, categorizing into the concepts Directory Service (DS) and Query Relay (QR), with our new Aggregating Discovery Service (ADS) approach. To do so, we elaborate on the different concepts, regarding the requirements, defined in Section 4. A summary of the comparison is depicted in Tables 1 and 2.

5.3.1 Directory Service (DS) approach

The Directory Service approach represents the most basic way to provide discovery service functionality. Given an EPC, a query to the discovery services would return a simple list of EPCIS server addresses that are in possession of read events for this particular EPC. Even though the design is simple, it has all means to provide functionality for the core requirements **RQ1** through **RQ4**. Discovery services are still question to research. However, security considerations such as for **RQ5**, **RQ8** and **RQ9** can be addressed by introducing existing authorization and authentication mechanisms, such as Public Key Infrastructures (PKIs). The integrity of the data (**RQ10**) can be ensured using digitally signed messages.

Data ownership (**RQ6**, **RQ7**) is a weak spot of this concept. The information residing at the discovery service comprises only of EPCIS server addresses. The actual read events are still stored at the respective EPCIS servers. However, according to [of Cambridge et al. (2007)] even such information is considered to be sensitive to some companies. To protect this information a discovery service following this concept would need to implement a role-based access layer. Such a layer is difficult to maintain, because of dynamic business relationships. Information about these business relations would need to be copied from the EPCIS servers to the discovery service, resulting in redundant information storage.

Business relationship independent design (**RQ11**) is directly related to the role-based access layer. Changing business relations are reflected by changing access permission for the particular trading partner. That means a company needs to update its access policies every time it adds, modifies, or removes permissions for trading partners.

Organic growth (**RQ12**) is a requirement that is hard to quantify in terms of good or bad. The DS concept provides a low technical boundary for potential users. However, due to the complex access control mechanism, it might be a problem for some companies to guarantee seamless collaboration with trading partners and at the same time protect their own interests. Frequently changing business relations have to result in frequent access policy updates at the discovery service. That is why we rate the support for organic growth for the DS concept rather small.

Looking at network traffic and produced data volume, we can state that the DS concept stores only a minimal set of data (EPC, EPCIS server address, timestamp). Messages between discovery service and client are also restricted to that type of information, leading to a small message size. Modern Database Management Systems (DBMS) are able to handle data volumes of many TB. The bigger problem is the potential request load, which increases with the number of clients and resources. These large data volumes produced by RFID enabled supply chains need to be searched very fast. This problem even aggravates when the number of parallel requests increases. We are currently conducting further research to analyze the impact of increased request load and data volume on the scalability of the different discovery service concepts. However, we expect the DS approach to be able to scale well by applying conventional scalability mechanisms such as load balancing and clustering. This assumption is based on the observation that most processing steps for the notification and the query of a discovery service can be parallelized very well.

RQ14 focuses on the quality of information. Assuming a suitable role-based access layer and a correct working query algorithm, the information returned by the discovery service is complete and correct.

We rate the client complexity (**RQ15**) for the DS concept high in comparison to the other two concepts. The DS approach only returns the URLs of the EPCIS servers' query interface. The client is responsible for invoking the individual EPCIS servers, to parallelize the different requests, to aggregate the information and to invoke successive request, related to different packaging hierarchies.

5.3.2 Query Relay (QR) approach

The Query Relay approach implements an asynchronous request/response paradigm, where the client submits a query for an EPC to the discovery service. The discovery service determines all potential resources for that EPC and propagates the query to these resources, which in turn answer directly to the client. The client needs to implement a callback interface, which is used to aggregate incoming EPCIS responses.

Just like the DS approach, the QR concept provides functionality for requirements **RQ1** through **RQ4**. An authorization and authentication layer needs to be implemented to restrict the number of authorized clients (**RQ5**).

Security (**RQ5**, **RQ8** and **RQ9**) can also be covered by introducing a PKI. By the same token, information integrity can be ensured using digital signatures based on certificates of PKI (**RQ9**).

In contrast to the DS approach, data ownership (**RQ6**, **RQ7**) is a strong feature of the QR concept. The discovery service relays the actual client query to the respective resources, which decide for themselves if they answer the request. That way, the resources are in full control of their data. The advantage is, that the discovery service does not need to provide a role-based access for the actual data. Redundancy and complex access right management are not necessary, because the responsibility to determine whether a client is allowed to receive the information is shifted to the resources.

Business relationship (**RQ11**) independent design is not as critical as for the DS approach, since the resources directly manage the access to their data. The interaction between client and discovery service is not affected by changing business relations. Clients need to negotiate with the operators of the resources, to get the desired information, but the discovery service is not involved in this process.

Organic growth (**RQ12**) is encouraged by the QR concept, because it requires less administrative overhead at the discovery service level to manage access policies. Therefore, it is easier for information providers and information consumers to join the network.

The discussion on scalability (**RQ13**) can be analogous to the DS approach in terms of network traffic and data volume. However, the QR approach has the advantage that it is relieved from complex role-based access policy checking, which can become a problem when the number of concurrent requests rises. So comparing the QR approach to the DS approach, the QR concept has a slight advantage.

Quality of information is a critical point in the QR concept. A client requesting information has no information about the number of potential EPCIS servers that are in possession of information regarding the queried EPC. Consequently, it has no information how many answers it needs to expect. A client querying a discovery service implemented according to the QR approach does not know how many answers from resources it has to expect. EPCIS servers might have slow response times, deny a response to his query or be temporarily not available. This leads to a situation that the client has to wait for a substantial amount of time to be sure that each EPCIS that replies to his query had the chance to do so. Therefore the client has to wait until a timeout is reached. This stands in contrast to a low response time. The result of a client query is complete and correct (**RQ14**) if and only if the client waits long enough to assure that no more replies are still underway. The client has no indication if EPCIS servers are temporarily unavailable.

The asynchronous communication inherent in the QR concept directly leads to an increased client complexity (**RQ15**). In the QR approach the client must be able to receive data from multiple previously unknown sources without knowing the exact number of responses. This results in the need for a complex software design that has to handle multiple incoming connections for a single request. Furthermore, the client has to aggregate the EPCIS responses by itself. Given the fact that client queries are forwarded to respective EPCIS servers immediately, the client is not in full control of its query. It cannot cancel the request or deny that his query is forwarded to a specific EPCIS, which might be a competitor's EPCIS.

5.3.3 Aggregating Discovery Service (ADS) approach

The ADS approach combines the advantages of the DR and the QR. The ADS shifts the complexity (**RQ15**) of query parallelization and the aggregation of EPCIS responses from the client to the discovery service and creates a view of the relevant information for the client. Hence, a query is immediately forwarded to all relevant EPCIS servers. The client is no longer in control of the query once it submitted it. If EPCIS severs enforce role-based access control this is not an issue because only the client role is revealed to the information holder, not the client identity.

Similar to the previous two approaches the ADS supports the four core functionalities (**RQ1**-**RQ4**). Security measures (**RQ5**, **RQ9**) and **RQ10**) can also be taken from the DS and the QR approaches.

The first major improvement compared to DS and QR is data ownership (**RQ6** and **RQ7**). The discovery services relays the client query to the respective EPCIS servers, providing complete privacy for the resources (**RQ8**), but in contrast to the QR approach, the ADS can control the query process, enabling it to take remedial action upon non-responding resources. The ADS is able to provide the client with a complete and correct set of information (**RQ14**), under the assumption that the client is allowed to see all the information. Otherwise, the result would contain a hint that there is additional information, the client is not allowed to see.

To show the scalability (**RQ13**) of our approach we discuss relevant aspects in Section 5.4.1. Since the ADS does not need to implement fine-grained role-based access control, to protect the companies' information, there is no need for adaptation when companies change their trading partners (**RQ11**). Closely related to this topic is the issue of organic growth. Low technical boundaries and flexibility regarding trading partner management encourage companies to add value to the EPC Network, to provide a beneficial environment for all participants. Table 1 and 2 summarize our evaluation of the presented discovery service concepts, regarding the requirements, defined in Section 4.

	Core Functionality				
	RQ1	RQ2	RQ3	RQ4	RQ5
DS	•	•	•	•	•
QR	•	•	•	•	•
ADS	•	•	•	•	•

Table 1. Fulfillment of Core Operational Requirements

	Selected Requirements											
	RQ6	RQ7	RQ8	RQ9	RQ10	RQ11	RQ12	RQ13	RQ14a	RQ15	RQ16	RQ17
DS	•	–	•	•	•	–	–	•	•	–	–	–
QR	•	•	•	•	•	•	•	•	–	–	–	–
ADS	•	•	•	•	•	•	•	•	•	•	–	–

Table 2. Fulfillment of Selected Requirements

5.4 Evaluation

One of the most important criteria for the successful operation of a discovery service for a larger community of collaborating trading partners is its ability to scale with an increasing number of participants. In detail, we need to focus on increasing network traffic, request load, and data volume. In this subsection, we draw a light on the scalability aspects of our ADS approach.

5.4.1 Scalability

The ADS provides additional functionality, which requires more computing power than the Directory Service or Query Relay approach. Like stated before the ADS has to wait for all responses of the subqueries, thus maintaining a connection's state for the request-response cycle with the client. In this section we show that it is possible to implement a scalable discovery service following the ADS approach.

We exemplify the potential load for a discovery service in the U.S. pharmaceutical supply chain by a back-of-the-envelope calculation. Following the supply chain network model of Williams et al. [Williams et al. (2008)] a discovery service has to deal with 1,000 notifications per second at peak times and 200 queries per second in average. We assume the worst case scenario that supply chain partners conduct a query for each item they notify as indicated by [of Cambridge & UK (2007)]. The ADS therefore has to deal with the same amount of queries to the discovery service. As the authors additionally state a supply chain does not exceed 15 partners.

5.4.2 Load balancing and data partitioning

Distributing incoming notification messages and client queries to many self-contained application servers allows the ADS to scale very well. HTTP load balancing can be performed in both, hardware and software for very high connection speeds. Additional servers can be added at any time allowing the system to grow in size.

HTTP reverse proxy servers balance incoming HTTP queries. They accept incoming HTTP connections and are able to act based on the queried URL or even arguments in the HTTP request. Implementations like the event-driven *nginx*[1] can help to lower the CPU load on application server machines by mapping requests to a specific EPC to one specific server. Each server is then responsible for a range of EPCs, implementing partitioning at the application server and database tier. Client queries always refer to one or more EPCs. No single database query will ever need to JOIN any data with rows for other EPCs. Database queries will only perform index lookups for EPCs and return the corresponding EPCIS URLs. This allows the database to scale by horizontally partitioning all data by EPC. Every database server is then responsible to serve requests for a range of EPCs. The database lookup only consists of small queries requiring basic database functionality. There is no need for complex locking mechanisms because all data has to be stored persistently and no tuples will be deleted or updated. Furthermore there is no need for synchronizing database partitions.

5.4.3 Open connections

As depicted above it is assumed that the Discovery Server has to handle about 1,000 client queries per second. For a supply chain with $n = 13$ enterprises in average this results in $\frac{n-1}{2} = 6$ relevant EPCIS servers in average per query. This results in 6,000 subqueries per second. Assuming a query to an EPCIS is replied to in one second on average the system has to hold about 6,000 connections simultaneously.

We tested how many connections one commodity PC is able to hold. To simulate a real-world scenario we requested 30,000 random Websites we gathered by querying a search engine with random keywords. All DNS resolving was done before starting the test at a limited upstream speed of 1 Mbit/s and 2.4 GHz CPU speed. Using asynchronous I/O processing we were able to have a single-threaded Python script sustain 3,000 connections (1,100 active) while using 22 to 24% CPU power. A low number of commodity-level servers can easily handle the total amount of connections.

[1] `http://nginx.net`

5.4.4 Bandwidth

In the basic 'Query Relay' architecture, the queried EPCIS servers reply directly to the client. In comparison, the ADS is the single response endpoint for all subqueries. Like described before, during peak hours the ADS has to be able to cope with 1,000 incoming client requests per second. For 6 relevant EPCIS servers on average, it has to send 6,000 subrequests and receive 6,000 subreponses per second. We expect each (sub)query to be 1 KB, each subresponse to be 2 KB in size, and each aggregated response to be 12 KB in size.
Receiving 1,000 queries/s at 1 KB per query and 6,000 subresponses/s at 2 KB per subresponse comes out to an inbound bandwidth of $\frac{(1{,}000 \cdot 1)+(6{,}000 \cdot 2) \cdot 8}{1000} = 104\ Mbit/s$. On the other hand, sending 6,000 subqueries/s at 1 KB per subquery and 1,000 aggregated responses/s at 12 KB per response equals an outbound bandwidth of $\frac{(6{,}000 \cdot 1)+(1{,}000 \cdot 12) \cdot 8}{1000} = 144\ Mbit/s$. Both throughputs are perfectly feasible using available internet connections.

5.4.5 XML handling

All replies sent back from EPCIS servers to the ADS use the XML format standardized by EPCglobal. It wraps all `ObjectEvent`s in a single `EventList` [EPCglobal (2007b)]. XML parsers optimized for high throughput provide efficient functionality for aggregating these XML responses. SAX or Pull parsers have proven their efficiency in SOAP environments where a large number of small XML queries have to be processed [Chiu et al. (2002)]. While receiving the XML data stream from a responding EPCIS every parsed tag inside the `EventList` can instantly be created on the output stream that, after all EPCIS servers replied, will be sent back to the client. This eliminates the need to add further buffers for XML objects and reduces XML rendering time.

6. Summary and future work

We started out by motivating the necessity of a discovery service for the EPC Network by introducing real world use cases that require the presence of such a component. In Section 3, we looked at the components of the EPC Network, discussed their particular roles within supply chain collaboration scenarios, and defined their relation to the discovery service. Section 4 introduced requirements for the implementation of a discovery service, followed by a description of existing theoretical and practical discovery service approaches. We also proposed a new design for an Aggregating discovery service, which we compared to the existing concepts in Section 5.
We see two major directions for future work. First of all, it is clear that there will not be a single discovery service, serving all industries. Scalability and political issues require to run a number of independent discovery services. Future research should include the investigation of inter discovery service communication and the definition of a communication protocol to support the exchange of information among independent discovery services. Secondly, we need to quantify the impact of an increasing number of clients onto a single discovery service or a network of interconnected discovery service, to support design decisions for the architecture of discovery services.

7. References

A. Rezafard, A. C. (2008). Extensible Supply-Chain Discovery Service Problem Statement, *IETF Proposal*.

Beier, S., Grandison, T., Kailing, K. & Rantzau, R. (2006). Discovery Services – Enabling RFID Traceability in EPCglobal Networks, *Proc. of the 13th International Conference on Management of Data (COMAD)* .

Chiu, K., Govindaraju, M. & Bramley, R. (2002). Investigating the Limits of SOAP Performance for Scientific Computing, *Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing* pp. 246 – 254.

EPCglobal (2007a). Architecture Framework Version 1.2.

EPCglobal (2007b). EPC Information Services Version 1.0.1.

EPCglobal (2008). Object Name Service Version 1.0.1.

EPCglobal (n.d.). Discovery Services Standard (under development).

Group, I. C. (2009). Ip crime report 2008-2009. IP Crime Report.

Ilic, A., A. Groessbauer and, F. M. & Fleisch, E. (2009). Understanding Data Volume Problems of RFID-enabled Supply Chains, *Business Process Management Journal*, Vol. 16.

Ilic, A., Groessbauer, A., Michahelles, F. & Fleisch, E. (2009). Estimating Data Volumes of RFID-enabled Supply Chains, *15th Americas Conference on Information Systems (AMCIS)*.

Kürschner, C., Condea, C. & Kasten…, O. (2008). Discovery Service Design in the EPCglobal Network, *The Internet of Things* .

Kürschner, C., Condea, C., Kasten, O. & Thiesse, F. (2008). Discovery service design in the EPCglobal network: towards full supply chain visibility, *IOT'08: Proceedings of the 1st international conference on The internet of things*, Springer-Verlag, Berlin, Heidelberg, pp. 19–34.

Melski, A., Müller, J., Zeier, A. & Schumann, M. (2008). Assessing the effects of enhanced supply chain visibility through rfid, *14th Americas Conference on Information Systems (AMCIS'08), Toronto, Canada*.

Müller, J., Oberst, J., Wehrmeyer, S., Witt, J. & Zeier…, A. (2009). An Aggregating Discovery Service for the EPCglobal Network, *hicss* .

Müller, J., Pöpke, C., Urbat, M., Zeier, A. & Plattner, H. (2009). A Simulation of the Pharmaceutical Supply Chain to Provide Realistic Test Data, *Advances in System Simulation, International Conference on* 0: 44–49.

OECD (2008). *The Economic Impact of Counterfeiting and Piracy*.

OECD (2009). *Magnitude of counterfeiting and piracy of tangible products*.

of Cambridge, A. U. & Research, S. (2007). High Level Design for Discovery Services. BRIDGE project.

of Cambridge, A. U. & UK, G. (2007). Requirements document of serial level lookup service for various industries. BRIDGE project.

of Cambridge, U., wireless, A., Research, B., Research, S., Zurich, E. & UK, G. (2007).

Ohnsman, A. & Kitamura, M. (2010). Toyota Recalls Increase on Brake Flaw Shared by Honda.

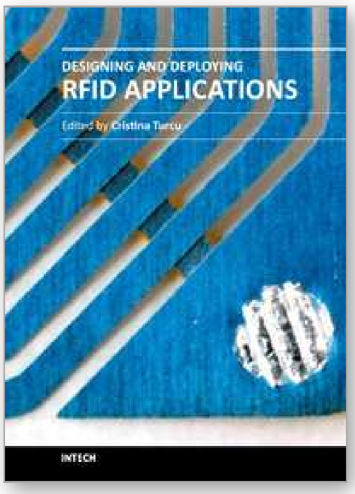
Polytarchos, E., Eliakis, S. & Bochtis, D. (2010). Evaluating Discovery Services Architectures in the Context of the Internet of Things, *Unique Radio Innovation* .

Rogers, E. M. (1995). *Diffusion of innovations*, Free Press, New York.

Shrestha, S., Kim, D. S., Lee, S. & Park, J. S. (2010). A Peer-to-Peer RFID Resolution Framework for Supply Chain Network, *Future Networks, International Conference on* 0: 318–322.

Simchi-Levi, D., Kaminsky, P. & Simchi-Levi, E. (2003). *Managing the Supply Chain : The Definitive Guide for the Business Professional*, McGraw-Hill.

Williams, J. R., Sanchez, A., Hofmann, P., Lin, T., Lipton, M. & Mantripragada, K. (2008). *Modeling Supply Chain Network Traffic*, p. 242.


DESIGNING AND DEPLOYING
RFID APPLICATIONS
Edited by Cristina Turcu
INTECH