Comparison of Discovery Service Architectures for the Internet of Things

Sergei Evdokimov, Benjamin Fabian, Steffen Kunz*
Institute of Information Systems
Humboldt-Universität zu Berlin
Spandauer Str. 1, 10178 Berlin, Germany
{evdokim,bfabian,steffen.kunz}@wiwi.hu-berlin.de

Nina Schoenemann*
Dpt. of Information Systems & Information Management
University of Cologne
Pohligstr. 1, 50969 Koeln, Germany
schoenemann@wim.uni-koeln.de

***Abstract*—In the emerging Internet of Things rich data on real-world objects and events will be generated in vast amounts and stored in widely distributed databases. In truly global and dynamic application scenarios, intermediate brokers are needed to find these data, even if the exact location and form of storage are initially unknown to the requester. Discovery Services are aimed to fill this gap: they respond to requests for data on specific objects with a list of corresponding data providers.**

In this paper, we frame functional requirements for Discovery Services, and perform an overview and analysis of five established approaches for implementing Discovery Services that are taken from literature and industrial practice. In order to compare their characteristics, we develop a quality framework based on literature review and an ISO standard for software quality.

I. Introduction

One of the main drivers behind RFID is its ability to efficiently provide data that can assist companies in optimizing their supply chain processes. According to IDTechEx [1], 1.97 billion RFID tags were sold in 2008 and 2.35 billion have been forecasted for 2009. Therefore, we believe there will be a fast growing demand for global-scale solutions for RFID-based supply chains. By integrating and transcending existing closed-loop applications, these solutions could constitute a cornerstone of the Internet of Things (IOT) [2].

Today's use cases in which RFID is applied for identification and tracking of objects are mostly confined to manufacturing or companies that only implement RFID together with selected supply chain partners. In the future IOT, data of real world objects and events will be available globally and in vast amounts. These data will be stored in widely distributed, heterogeneous information systems, and will also be in high demand by business and end user applications. Therefore, a discovery mechanism that allows accessing such data is needed, even if its location and form of storage are unknown to the requester. It is conceived that so called Discovery Services (DS) will respond to such requests by returning a list of corresponding data providers [3].

Regarding the design of DS, there are several design decisions to be made. In this paper, we will present five important approaches that propose distinct architectures for DS, and compare their characteristics. Qualitative attributes for DS architectures have not been comprehensively studied in research so far. The novelty of those architectures, and the fact that most of them are currently only available as architecture proposals or pilot implementations, makes a quantitate evaluation and benchmarking difficult, if not impossible. Therefore, we construct a quality evaluation framework for conducting an evaluation of the existing DS architectures.

Our paper is structured as follows. In section II, we collect basic functional requirements for DS, which help to define our focus of discourse. In section III, we provide brief descriptions of existing DS architectures. In section IV, we compare and evaluate these architectures. For this purpose, we build a quality framework that is derived from literature and the ISO/IEC 9126 software quality evaluation standard. The framework is then used as a basis for a structural comparison of the DS architectures. In section V, we summarize our findings and give an outlook on further research.

II. Functional Requirements

In the following, we present a definition for an IOT DS based on the functional requirements it has to fulfill. Though not uncontested in requirements research, we adopt the practically established distinction between functional and nonfunctional requirements [4, ch. 6]. Functional requirements describe the functionality and services that a system should provide. Non-functional requirements often consider qualities or constraints on the system functionality, such as performance and installability, among others.

Important functional requirements for DS can be collected from the documents of EPCglobal, e.g., the ONS specification [5]. Relevant further work on requirements elicitation for DS has been conducted by the BRIDGE project together with GS1, including interviews with companies [6]. The requirements collected focus on functional and performance aspects, availability, integrity, as well as provider data confidentiality, but are rather neglecting the client's perspective.

Another recent line of research on DS is presented in [7], where similar requirements to those presented here are identified. Afilias Inc. formulated issues [8, p. 4] that build the design goals for its DS. Both mainly non-functional

*All authors contributed equally to this work and are therefore given in alphabetical order.

requirement sets are primarily used as arguments for the DS architecture chosen by [7] and [8]. While presenting a peer-to-peer alternative to the Domain Name System (DNS), [9] have collected a short set of functional, performance, and robustness requirements for general name services, which are also relevant for globally operating DS. The main source we used in this section is the compilation and discussion of IOT name service requirements given by [10].

To formulate the general requirements in an objective way, we will use the term *OID* (Object Identifier) in the following section, since a DS should not be limited to serve only EPC (Electronic Product Code) numbering schemes, but also arbitrary current or future object numbering systems. Based on the cited literature and conceptual analysis, the following set of high-level functional requirements for a DS can be identified:

1) *Flexible OID Support*: DS should be flexible in its support for different OID schemes.
2) *Publishing*: A data provider shall be able to input address documents into DS for OIDs for which he is authorized to publish data. Those documents shall include addresses of servers for EPC Information Services (EPCIS), which provide data about the objects carrying those OIDs.
3) *Multiple Publishers*: Multiple, independent, but authorized publishers should be able to provide data for an OID by storing corresponding address data in DS.
4) *OID Querying*: On an input of an OID by a client, DS shall output a current list of servers offering data about the object corresponding to the OID.
5) (Optional:) *Attribute Querying*: On input of some object attributes by a client, DS shall output a current list of servers offering information about objects matching the search attribute.
6) *Updating*: Authorized publishers shall be able to update the data records they published at will.
7) *Deleting*: Authorized publishers shall be able to delete the data records they published at will. A time-to-live value (TTL) should be provided for each document to indicate old data and to reduce overhead for deletion.
8) *Class-level Addresses*: If the OID is structured into a class-level and serial-level part, DS shall be able to work with partial OIDs at the class-level.
9) *Serial-level Addresses*: If the OID is structured into a class-level and serial-level part, DS should be able to work with fully serialized OIDs, for example a complete SGTIN EPC consisting of EPC Manager, Object Class, and Serial Number.
10) (Optional:) *Object Data*: To reduce query overhead, given an OID DS should be able to directly provide some partial object data. E.g., a DS can indicate that an object's official lifetime has expired.
11) (Optional:) *System Membership and Authorization Procedure*: A set of membership definition and organizational as well as technical authorization procedures for all publishers and clients of DS shall be provided.

We use the set of non-optional functional requirements to define core requirements a DS has to fulfill in today's business scenarios. Optional requirements focus on additional aspects that may become important in future applications, but should already be anticipated today. The core functional requirements are also useful to separate DS from related systems. E.g., the Object Name Service (ONS) [5] of EPCglobal does not fulfill DS functions according to our definition, since requirements 3) and 9) are not satisfied. ONS allows only one publisher, the EPC Manager, and supports only class-level information lookup.

III. Existing Architectures

In this section, we describe existing architectures for implementing DS.

A. EPCglobal

The concept of DS was first formulated by EPCglobal in [11]. The EPCglobal standard that should define the architecture and interfaces of the DS has not been published yet.[1] Available literature (see [3], [7], [12], [13]), however, provides a high-level description of the EPCglobal DS architecture that is depicted in Figure 1. The EPCglobal DS is organized as a lookup service that stores references (URIs) to EPCIS linked to EPC numbers about which they provide data. Such service will allow a client who is looking for data related to a specific EPC number to identify EPCIS that can provide these data.

The following steps describe the process of data exchange between the client, the DS, and corresponding EPCIS: In step (0a-b), multiple companies notify the DS about events that were recently added to their EPCIS repositories. (1) The client wants to get data about all events that are related to a given EPC number and issues a corresponding query to the DS, receiving the references to EPCIS that hold any events related to the queried EPC number. Then, (2) the client sends these references to the to the Domain Name Systems (DNS), which resolves the references into IP addresses. Using these IP addresses, (3a-b) the client queries the EPCIS that can provide data about the corresponding EPC.

It is assumed that the described architecture will provide a *local* discovery that enables participants within one or among several supply chains to share EPC data. Initially, the supply chain participants may not be aware of the EPCIS of others, but interfaces of the DS (e.g., its URL) have to be known in advance. It is, however, also possible that some items leave the "premises" of a given DS, but the information about related EPC events may still have to be made accessible to the parties that use other DS instances. According to EPCglobal, DS will constitute one of the EPCglobal Core Services [3] – services that will be operated by EPCglobal or its delegates. The actual role of EPCglobal is not yet defined. One of the possibilities is that EPCglobal will be providing services that enable interoperability of independent DS (steps 4, 5a-b). Available documents do not reveal any details on how such *global (inter supply chain)* discovery will be implemented.

[1] http://www.epcglobalinc.org/standards/

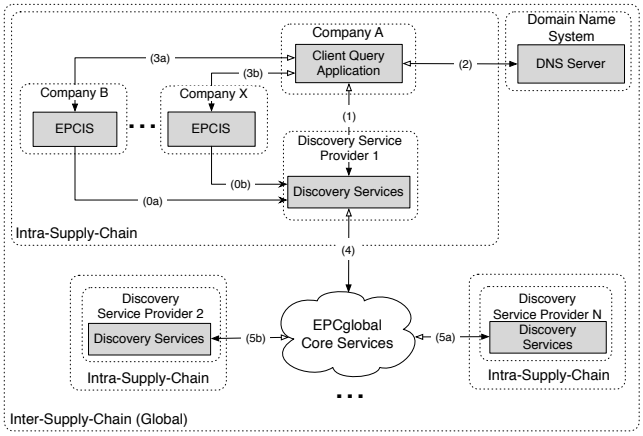

Fig. 1. EPCglobal Discovery Services

B. BRIDGE Project

The BRIDGE project, supported by the EU and coordinated by GS1, addresses a wide spectrum of problems related to the implementation of RFID in Europe. [14] provides high-level descriptions and analysis of a number of approaches that could be used for implementing DS. According to the BRIDGE vision, the DS can be implemented as directory services, and deployed either as a single server or as a network of federated servers that provide serial-level lookup for EPCIS containing data about objects identified by EPC numbers. The following four architectures were considered by BRIDGE.

Directory of Resources: In this approach, resources (EPCIS) register information about the availability of data corresponding to EPC numbers at the DS. The clients query the DS for obtaining references to the EPCIS that contain data about EPC numbers of their interest. Afterwards they query corresponding EPCIS for detailed data.

Notification of Resources: In this architecture model, clients first subscribe to the DS for a notification about certain EPC numbers. An EPCIS transmits information about the set of EPC numbers it holds data about. The DS then sends notifications about the availability of data for the EPC numbers of their interest to the clients. To obtain detailed data, the clients query the corresponding EPCIS.

Notification of Clients: EPCIS publish information about the availability of data for certain EPC numbers to the DS. The clients, on the other hand, use the DS for notifying resources about their interest in particular EPC numbers. If information about these EPC numbers is updated, the resources inform the clients, and the clients contact the associated EPCIS for detailed information.

Query Propagation: EPCIS publish information about the availability of data corresponding to EPC numbers to the DS. Clients use the DS for forwarding their queries directly to resources containing information about EPC numbers of their interest. The resources respond with detailed information.

For implementation of the proposed models, components such as LDAP, DNS, DHT or search engines are considered. As a proof of concept, a DS prototype was implemented [15], which will herein be referred to as the reference architecture of the BRIDGE project. The prototype is based on the Directory of Resources model with the storage component implemented as LDAP. The high-level architecture of the prototype is identical to the intra-supply-chain part of the EPCglobal DS architecture (Figure 1).

C. Afilias

Afilias Discovery Services were developed by Affilias Inc. and aim at solving five main issues in the IOT [16]: (a) unique identification of items in a world of diverse identifier authorities, (b) backward compatibility with existing identification schemes, (c) concerns regarding control of a single point of authority that is outside local boundaries, (d) assurance of practicality, scalability and openness to competition in the provision of services, and (e) trust / security of the system.

For supporting multiple, independent supply chains, Afilias DS utilize an open, Web services protocol called the Extensible Supply-chain Discovery Service (ESDS) [17]. ESDS provides description of concepts, the schema, and commands for implementation of ESDS Client Query Applications. A specification of ESDS was submitted to the IETF standardization committee in 2007. Some pilot projects have been conducted with partnering companies, e.g., in the air transportation industry. For participants of the still running pilot program, Afilias provides the DS infrastructure, technical support and a toolkit for the development of a client interface.

Afilias DS are based on DNS and are compliant to the architecture framework of EPCglobal. Identifier authorities can set up naming systems under existing top or country level domains (like ".org"). A supply chain with an identifier authority can thus establish naming system operations without a third-party identity that controls the global naming system. The identifier authority has to adopt a translation mechanism (such as EPCglobal has done with ONS) to translate their identifiers for DNS compatibility. Basic characteristics of Afilias DS are hierarchical lookups and DNS-based naming and translation.

Figure 2 shows the architecture and the discovery procedure with ESDS. Each participating supply chain has to instantiate an ESDS Server that handles publication of events (0a-b) and service discovery requests (1) within the supply chain. These are usually followed by a DNS resolution (2) and EPCIS access (3a-b). If a client requests external data, the ESDS server sends a global lookup (4) to the Afilias DS. This lookup is routed to other ESDS servers (5). Any local ESDS server will respond to incoming global lookups (6a-b), if it has the requested data. According to the ESDS internet draft [17, p.23], the discovery of services outside a supply chain will utilize a peer-to-peer protocol such as JXTA, but if and how this will be implemented is not made public by Afilias. Until now, the system depends on centrally hosted registries. Furthermore, ESDS specifies security mechanisms

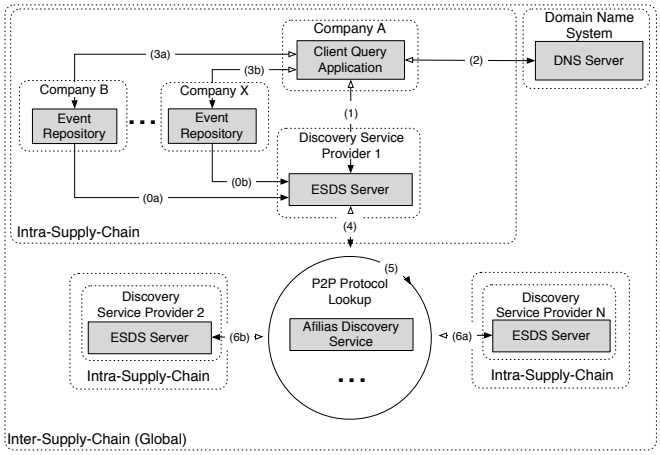

Fig. 2. Afilias Discovery Services

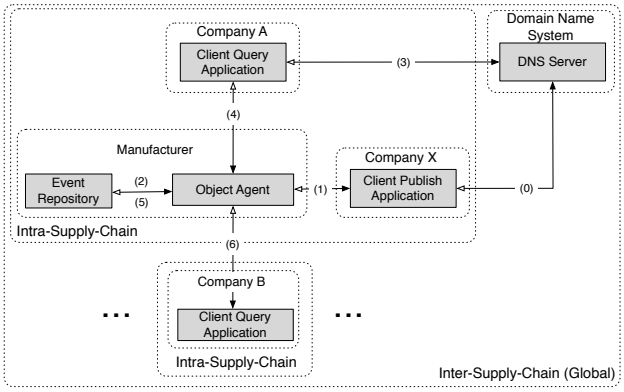

Fig. 3. ID@URI Discovery Service

for authenticated access to local data. Every participating company can define individual access rights for its own data.

D. ID@URI

The ID@URI naming system, also referred to as the DIALOG system, was developed in the context of the Dialog project[2] that started in 2003 as an open source project with the aim to develop a worldwide tracking and tracing system [18]. In the Traser EU project,[3] focusing on the tracking and tracing of individual items between small enterprises, and the Promise EU project[4] that focused on Product Lifecycle Information Management (PLIM) [18], the development of the ID@URI naming system was continued.

The ID@URI system was designed to use existing naming standards allowing interoperability and smooth integration with existing information systems. It has two core components: Client and Object Agent. The Client is used for reading product identifiers and connecting to the Object Agent identified by the ID@URI object identifier. While the URI part is the domain name of the Object Agent belonging to the company that manufactured the object, the ID part is unique within the address space of the URI. The ID is assigned locally at the manufacturing company when the object is created. For the ID, any naming system could be applied, e.g., the EPC, Global Trade Identification Number (GTIN), existing serial numbers, etc. [18].

Due to the fact that all data about a product is managed by one Object Agent and is stored in its domain, only one source of data has to be discovered. The actual discovery process functions as follows: In step (0), a Client Application that wants to publish data about an object ID-related event first resolves the URI of the object to the IP address of the corresponding Object Agent; then, (1) the Object Agent is contacted by the Client and (2) the event data is stored in the corresponding local Event Repository. When a Client wants to access events that refer to an object identified by some ID, (3) a querying application resolves the URI to an IP address and (4) connects to the Object Agent, which (5) uses the object ID for retrieving the data from the Event Repository to send it back to the Client Query Application. Inter supply chain queries (6), i.e., requests from clients in other supply chains, are managed in the same way as intra supply chain queries [18].

E. Peer-to-Peer Approaches (DHT-P2P)

Peer-to-Peer Systems (P2P) are highly distributed alternatives to classical network service architectures and can be considered to be a paradigm shift from the classical client–server architecture to a new paradigm with a roughly equal distribution of responsibility and load among peers. Especially structured P2P systems using Distributed Hash Tables (DHT), offer high robustness to faults, avoid single points of failures (e.g., they have no special root nodes like DNS), and distribute responsibility and load among participants in a systematic way by means of a prearranged topological overlay structure [19].

Since DHTs offer fundamental lookup functionality of arbitrary identifiers (e.g., an EPC) to nearly arbitrary answer documents (e.g., one or multiple EPCIS addresses), they can also constitute a foundation for highly scalable and robust global DS. This has been proposed mostly in academic literature so far, see, e.g., [20] and [21]. In [22] an implementation of a DHT-based DS called OIDA has been presented, which is based on the Bamboo DHT and was tested on roughly 350 globally distributed nodes of the experimental platform PlanetLab. In [21], simulation results (using the Pastry DHT) showed the feasibility for P2P networks consisting of up to 20,000 nodes. For DHT scalability with even higher node counts, theoretical performance results (see, e.g., [19]) or real-world applications (like the Vuze DHT[5]) can be consulted.

[2]http://dialog.hut.fi/
[3]http://www.traser-project.eu/
[4]http://www.promise-plm.com/
[5]http://azureus.sourceforge.net/

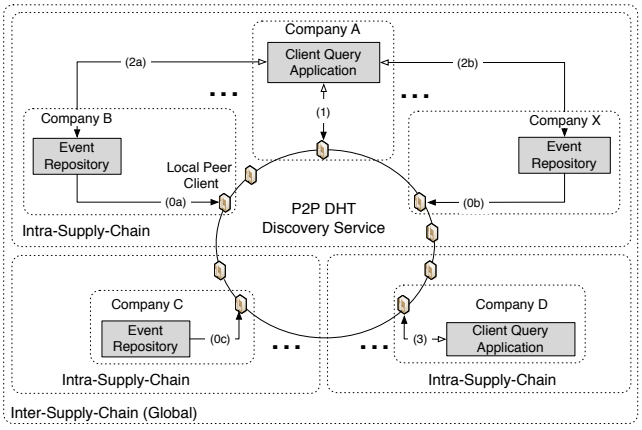

Fig. 4. P2P Object-Data Discovery

Fig. 4 shows how a DHT-based DS can be integrated into an EPCglobal-compliant application landscape. Once an EPC Capture Application at a partner company has detected an event involving an EPC, corresponding data can be published using an Event Repository (0a-c). To enable other partners to discover the EPCIS, a Local Publish Application (here Event Repository) uses a Local Peer Client to insert a pair (key=*h(EPC)*, value=*EPCIS address document*) into the DHT system, where h denounces a (cryptographic) hash function like SHA-1. A Client Query Application, which wants to discover data on a given EPC, can use the DHT Lookup Interface via its own Local Peer Client to get the IP addresses of the corresponding Event Repositories (1) and retrieve all data stored for an EPC (2). Inter supply chain EPC requests (3) function in the same way as intra supply chain requests.

F. Others

One of the first approaches to harmonize Web Service DS is the *Universal Description, Discovery and Integration (UDDI)* [23]. It depends on registry providers, who publicly offer UDDI services. UDDI was not designed for IOT-DS and lacks appropriate scalability. Moreover, since the shutdown of major UDDI registry providers (IBM, Microsoft, and SAP), it has fallen into oblivion.

A promising approach was *World Wide Article Information (WWAI)* developed by Stockway. The approach is based on a P2P protocol and has some features of the described DHT-P2P architecture. First announced as an open system, it is now a commercial tracking software.

The last to mention is the *Ubiquitous ID Technology* of the uID Center in Japan [24]. This center operates resolution server and database as well as an authentication authority (eTRON). It provides a very centralized approach, concurrent to EPCglobal, where the uID Center is in control of the infrastructure and specification development.

IV. Comparison of the Architectures

In this section, the scope and quality of the DS approaches are compared. For scope, we consider high-level characteristics that describe goals and the status of the corresponding projects. The quality evaluation concentrates on the technical aspects of the approaches and is based on the ISO/IEC 9126 standard [25] for the evaluation of software quality, review of the relevant literature, and interviews with subject area experts.

A. Scope

Before comparing the DS architectures in detail, we focus on their general characteristics. Our findings are presented in Table I. The individual items are described in the following.

In order to be able to analyze and compare the architectures, at least their *high-level descriptions* should be available. Not all of the considered approaches, however, are already provided with *detailed architectures*. While EPCglobal lacks this detailed description, for the others listed here these descriptions are available. They include details on data schemes and overlay communication protocols (i.e., ESDS in case of Afilias, and DHT in case of DHT-P2P approaches).

Standardization describes the normativeness of the approaches. EPCglobal is a clear leader in this aspect. Major parts of its architecture framework are *standardized* by ISO (excluding the DS so far, of course). Beyond, only Afilias is trying to standardize ESDS by IETF. All of the approaches are designed to *support the EPC standards*, apparently this is due to EPCglobal's prevalence. This means that the EPC can be used as a numbering scheme and EPCIS can be integrated into the system architecture.

A DS has a *global scale*, if it is designed to support data exchange between participants that do not necessarily have business partnerships with each other. The BRIDGE project aims at providing discovery for stakeholders that are limited to a supply chain and for which client applications are preconfigured with entry points of the corresponding instance of a DS. EPCglobal mentions that, at least partially, the DS will be a part of the EPCglobal Core Services and will be operated by EPCglobal or its delegates. This suggests that EPCglobal plans to support global inter supply chain discovery by providing a service responsible for intercommunication between local DS instances. Afilias also considers local and global discovery. ID@URI and DHT-P2P approaches were specifically designed for enabling global discovery.

State captures the maturity of projects and other activities regarding the development of the DS architectures.

Business model describes commercial constituents of the projects. BRIDGE, ID@URI, and DHT-P2P are organized as research initiatives. Afilias is planning to offer payed hosting services for ESDS, while EPCglobal has not yet released any statements regarding membership fees or availability of any payed services. So far, the EPCglobal business model for EPC naming and ONS is based on membership fees.

Dedicated infrastructure describes which dedicated services (including hardware and software) are needed by the proposed approaches. In case of EPCglobal, there will be relevant core services provided by EPCglobal or its delegates as well as local DS instances. As for BRIDGE, no inter supply chain communication is provided, only local DS instances are

TABLE I
COMPARISON OF SCOPES

	EPCglobal	**BRIDGE**	**Afilias**	**ID@URI**	**DHT-P2P**
High-level description	yes	yes	yes	yes	yes
Detailed architecture	no	yes	yes	yes	yes
Standardization	partly	no	in progress	no	no
Support of EPC	yes	yes	yes	yes	yes
Global scale	not available	no	yes	yes	yes
State	the standard is in *To Be Developed* state	software pilot, the project is closed	pilot phase	software is available	in progress
Business model	not available yet	research project	registry hosting (planned)	research project, open source	research project
Dedicated infrastructure	core services provided by EPCglobal, DS instances	DS instance (distributed if implemented as P2P)	any registry hoster	event repository hosted by manufacturer	distributed across stakeholders
Power agglomeration by	hosters of local DS instances, EPCglobal or its delegates	hosters of DS instances, distributed if implemented as P2P	hosters of local DS instances, Afilias	manufacturer	none

required. In case of Afilias, these are instances of ESDS. It may also be possible that peer-to-peer communication between ESDS instances will rely on certain services provided by Afilias. As far as ID@URI is concerned, the core infrastructure is hosted by the manufacturer of the tagged product.

Power agglomeration describes entities that can obtain a certain level of control over the DS. Thus, in cases where a DS instance is implemented as a centrally hosted directory, the entity responsible for the hosting is in position to decide about how the service will be provided to the stakeholders. In case any centrally managed services are required, the provider of these services can also be considered as being in (partial) control over the DS. On the other hand, a P2P-based solution could ensure a distributed infrastructure where no single instance (neither participant nor provider) has superior power.

B. Quality

In order to assess the quality of the different DS architecture approaches, we conducted a literature analysis of important quality categories. We identified three major sources for defining the quality of DS. First, the ISO/IEC 9126 [25] standard defines a quality model applicable to any kind of software. It consists of six main quality categories with corresponding subcategories - given in brackets: *Functionality* (suitability, accurateness, interoperability, compliance, and security), *reliability* (maturity, fault tolerance, and recoverability), *usability* (understandability, learnability, and operability), *efficiency* (time behavior and resource behavior), *maintainability* (analyzability, changeability, stability, and testability), and *portability* (adaptability, installability, conformance, and replaceability).

Second, based on interviews with experts and end users, literature review and project work, [7] identified the following six requirement categories for DS: *data ownership*, *security*, *business relationship independent design*, *organic growth*, *scalability*, and *quality of service*. For each of these categories, the authors defined corresponding requirement hypotheses leading to a total of eight hypotheses on requirements.

Third, the following subcategories were specified in the BRIDGE project [14]: *horizontal scalability*, *bottleneck*, *data update*, *data search*, *organization of data*, *record with fields*, and *guarantee of result correctness*.

We consolidated the three different approaches with ISO/IEC 9126 as the basic framework. The resulting DS quality framework is depicted in Table II. In particular, we left out the following ISO/IEC 9126 subcategories, because they could not be applied to the DS architectures – mainly due to the fact that we are currently only able to compare high-level architecture concepts: *stability*, *testability*, *adaptability*, *conformance*, and *compliance*. We added the three subcategories *privacy*, *access control*, and *trackability* taken from [7] and [14]. All other requirements of [7] and [14] could be mapped to the ISO categories. In the category *reliability*, we subsumed all ISO subcategories into one, because they were strongly related to each other. For the category *suitability* we evaluate whether the DS architectures fulfill the functional requirements presented in section II. Furthermore, we changed the name of *replaceability* to *expandability*, because this is more relevant for our subject of study.

The fact that complete descriptions of most of the DS approaches are currently not available makes it difficult to evaluate and benchmark their quality. Therefore, we decided to do an expert evaluation for each of the approaches and then discuss and triangulate the individual evaluation results in an expert group discussion. We applied our quality framework to the approaches of EPCglobal, Afilias, ID@URI, and DHT-P2P. Due to the similarity to the approach of EPCglobal, the BRIDGE architecture prototype was excluded from the quality evaluation. As BRIDGE was involved in the EPCglobal Data Discovery Joint Requirements Group, it can also be assumed that parts of its solution will be incorporated into the EPCglobal DS standard. The quality evaluation results are presented in Table II, which describes how the approaches address the considered quality categories.

As far as suitability is concerned – subsuming the core (non-optional) requirements presented in section II – this quality is fulfilled by all considered approaches. In case of the optional functional requirement *attribute querying*, none of the current DS designs appear to offer this functionality, though for certain

TABLE II
DISCOVERY SERVICE QUALITY COMPARISON

Quality Categories	ISO	[7]	[6]	Description	EPC-global	Afilias	ID@URI	DHT-P2P
Functionality								
Suitability	x			The DS fulfills the functional requirements	+	+	+	+
Accurateness	x	x		The DS query result is complete and correct.	o	o	o	o
Interoperability	x	x		The DS architecture encourages participation by providing open interfaces and specifications.	+	+	+	+
Security	x	x		The DS protects the confidentiality, integrity and availability of published and queried data.	o	o	o	+
Privacy			x	The DS ensures and protects client privacy.	-	-	-	+
Access control		x		Information publisher are able to define and control access rights.	o	o	-	o
Trackability		x		Information publisher are able to track requests upon their data as well as its usage.	o	o	-	-
Reliability								
Fault tolerance, recoverability	x	x		Even if parts of the overall system are not functional, the DS can assure a certain quality of service.	o	o	o	+
Efficiency								
Communication scalability	x	x	x	The DS does not have bottlenecks when carrying out multiple data update and search operations.	-	-	-	+
Resource scalability	x	x	x	New participants can easily join the DS. There is no limitation to the number of participants.	+	+	+	+
Maintainability								
Analyzability	x			Failures in the DS can be detected and repaired, configuration can be adjusted.	+	+	+	-
Changeability	x	x		Changes in business relationships shall not affect the way in which a company interacts with the DS.	o	o	o	+
Portability								
Installability	x			The DS can easily be integrated into existing information systems.	+	+	+	+
Conformance	x			The DS conforms with existing exchange and naming standards.	+	+	+	+
Expandability	x			The design of the DS is open for extensions.	+	+	+	+

(Quality evaluation: + good, o concerns, - poor)

information queries in the future IOT this functionality could become very important (e.g., to query for data on all large and heavy objects that are on the way to a certain location). Currently, only the DHT-P2P approach is known to be able to fulfill the optional requirement *object data*, i.e., caching small amounts of object data to reduce query overhead. In the IOT, this functionality could help to improve performance issues. As fas as the requirement *system membership* and authorization procedure is concerned, none of the proposals discusses this in detail, though it is a major precondition for implementing security functionality.

None of the considered approaches can guarantee *accurateness* of query results, because the event data is provided by a third party that can omit or fabricate some values, or just deny access if the requested information is too sensitive. Another category that is naturally characteristic to DS and therefore fulfilled by all approaches is their aim at providing *interoperability*, – i.e., in each case, descriptions, interfaces, and specifications are made publicly available.

In terms of *security*, except for DHT-P2P, no other approaches (at least at the current stage) discuss detailed measures aimed to preserve confidentiality and integrity of the stored and transmitted data and ensuring availability of the related services. The same is true for *privacy* of a client issuing a query to the DS – neither EPCglobal, Afilias, nor ID@URI approaches discuss any mechanism that could prevent DS provider from identifying the client (e.g., its IP address). In contrast to that, the architecture of the P2P-DHT approach naturally conceals the client, as a query typically reaches its destination in more than one hop in the P2P overlay. All four approaches assume that the publishers are able to define *access control* rights for their data. Saying "data", we mean here data about events as well as entries published to the DS repository. Since the DS are not (physically) hosted by the stakeholders, their control over the entries published to the DS is limited. However, in EPCglobal, Afilias, and primarily also for DHT-P2P approaches it is assumed that the actual events are stored in the repositories controlled by the stakeholders. Only in the ID@URI approach, the event data has to be transferred to a third party – the manufacturer.

Concerning *trackability*, approaches of EPCglobal and Afilias assume that stakeholders are able to track access to their data including usage. In the ID@URI case, it is assumed that the information is always provided by the manufacturer who has no direct incentives to provide tracking information. In the P2P-DHT case the trackability becomes even a bigger issue, as the DS are represented by thousands of peers – many not even being aware of each other. Concerning *reliability* and *efficiency*, the approaches that assume that DS instances are deployed centrally, inevitably induce a bottleneck. Measures as replication and load-balancing can improve *fault tolerance*, *recoverability*, and *communication scalability*. However, it is only the DHT-P2P-based approach that removes the bottleneck completely and introduces mechanisms that ensure that the DS remains functional if (major) parts of the underlying infrastructure become unavailable. At the same time, none of

the approaches imposes any explicit restrictions on the number of maximum participants, what could restrict *participant scalability*.

The *analyzability* and *changeability* categories describe how easy the system is to maintain. The centralized nature of EPCglobal, Afilias, and ID@URI in principle allows to localize and identify errors and failures. On the other hand, the architecture of DHT-P2P, with its much more flexible structure, can make localizing failures a difficult task. At the same time, such flexibility gives this model an advantage when the business relationships of stakeholders are changed and these changes affect data flows between them. With P2P, the changes are naturally propagated through the whole overlay. The other approaches can require stakeholders updating several DS instances or, in certain cases, even a deployment of new DS instances. Regarding *installability*, ease of deployment is one of the prerequisites of all the approaches. The same can be said about *conformance* and *expandability*.

V. Conclusion

In this article, we described requirements and analyzed the following five approaches for implementing DS in the IOT – EPCglobal, BRIDGE, Afilias DS, ID@URI, and DHT-P2P. For comparing and evaluating these innovative architectures, we developed a quality framework based on the ISO/IEC 9126 standard and a literature review. This framework does not only provide a structured analysis tool, it can also be used by software developers, consulting companies, and service providers to individually evaluate different solutions for DS in the IOT, enabling a deeper understanding, improvement, or mutual integration of the approaches. Subsuming the current state of DS designs, it can be stated that the EPCglobal approach is still in development. Components of the relatively mature Afilias approach will probably be integrated into the EPCglobal standard, while the prototype of the BRIDGE project is very similar to the currently discernible EPCglobal approach. All three of them share the same advantages and disadvantages. Two very contrasting approaches are ID@URI and DHT-P2P. Both have several advantages and disadvantages: ID@URI is easy to deploy, but is dependent on the manufacturer who alone has the responsibility for providing object information in the supply chain. DHT-P2P delivers a very scalable and flexible solution. However, it remains unclear, who should bootstrap such an approach and how support for trackability and analyzability can be improved.

Limitations of our work are primarily the lack of real-world tests. The current lack of detailed architecture definitions also restrict a sound analysis, but as soon as, e.g., EPCglobal, has specified its DS standard and more implementations are available, the analysis could be resumed. The quality framework will provide an appropriate structure and starting point for further research. More work should be conducted on really independent and global scale architectures apart from EPCglobal, fostering the development and acceptance of the Internet of Things.

References

[1] R. Das and P. Harrop, "RFID Market Forecasts 2009-2019," IDTechEx, Tech. Rep., Apr. 21 2009.

[2] N. Gershenfeld, R. Krikorian, and D. Cohen, "The Internet of Things," *Scientific American*, vol. 291, no. 4, pp. 76–81, Oct. 2004.

[3] EPCglobal, "The EPCglobal Architecture Framework – Version 1.3," March 2009. [Online]. Available: http://www.epcglobalinc.org/standards/architecture/

[4] I. Sommerville, *Software Engineering*, 7th ed. Addison Wesley, 2004.

[5] EPCglobal, "EPCglobal Object Naming Service (ONS) 1.0.1," EPCglobal, May 2008. [Online]. Available: http://www.epcglobalinc.org/standards/ons

[6] BRIDGE, "BRIDGE WP02 – Requirements Document of Serial Level Lookup Service for various Industries," August 2007.

[7] C. Kürschner, C. Condea, O. Kasten, and F. Thiesse, "Discovery Service Design in the EPCglobal Network – Towards Full Supply Chain Visibility," in *Proceedings Internet of Things (IOT 2008), Zurich, Switzerland, 2008*, ser. LNCS 4952. Springer-Verlag, Berlin-Heidelberg, 2008, pp. 19–34.

[8] Afilias, "Finding your Way in the Internet of Things," White Paper, Sep. 2008. [Online]. Available: http://www.afilias.info/webfm_send/11

[9] V. Ramasubramanian and E. G. Sirer, "The Design and Implementation of a Next Generation Name Service for the Internet," in *Proceedings ACM SIGCOMM '04, Portland, Oregon, USA*, 2004, pp. 331–342.

[10] B. Fabian, "Secure Name Services for the Internet of Things," Ph.D. dissertation, Humboldt-Universität zu Berlin, Wirtschaftswissenschaftliche Fakultät, 2008. [Online]. Available: http://edoc.hu-berlin.de/docviews/abstract.php?id=29312

[11] EPCglobal, "The EPCglobal Architecture Framework – Version 1.0," March 2005. [Online]. Available: http://www.epcglobalinc.org/standards/architecture/architecture_1_0-framework-20050701.pdf

[12] GS1 Germany, "Internet der Dinge. Management Information. Das EPCglobal Netzwerk," Tech. Rep., September 2005.

[13] Verisign, "The EPC Network: Enhancing the Supply Chain," January 2004. [Online]. Available: http://www.verisign.com/static/002109.pdf

[14] BRIDGE, "BRIDGE WP02 – High Level Design for Discovery Services," August 2007.

[15] ——, "BRIDGE WP02 – Working Prototype of Serial-level Lookup Service," February 2008.

[16] Afilias, "Afilias Discovery Services: Enabling Secure, Selective Visibility in Global Supply Chains. White Paper," White Paper, 2008. [Online]. Available: http://www.afilias.info/webfm_send/37

[17] A. Rezafard, "Extensible Supply-chain Discovery Service Problem Statement," IETF Internet-Draft, Nov. 17 2008, draft-rezafard-esds-problem-statement-03. [Online]. Available: http://tools.ietf.org/html/draft-rezafard-esds-problem-statement-03

[18] K. Främling, M. Harrison, and J. Brusey, "Globally Unique Product Identifiers – Requirements and Solutions to Product Lifecycle Management," in *Proceedings of 12th IFAC Symposium on Information Control Problems in Manufacturing (INCOM)*, 2006, pp. 17–19.

[19] H. Balakrishnan, M. F. Kaashoek, D. R. Karger, R. Morris, and I. Stoica, "Looking up Data in P2P Systems," *Communications of the ACM*, vol. 46, no. 2, pp. 43–48, 2003.

[20] B. Fabian and O. Günther, "Distributed ONS and Its Impact on Privacy," in *Proceedings IEEE International Conference on Communications (IEEE ICC 2007), Glasgow*, 2007.

[21] N. Schönemann, K. Fischbach, and D. Schoder, "P2P Architecture for Ubiquitous Supply Chains," in *17th European Conference on Information Systems (ECIS'09), Verona, Italy*, 2009.

[22] B. Fabian, "Implementing Secure P2P-ONS," in *Proceedings IEEE International Conference on Communications (IEEE ICC 2009), Dresden*, 2009.

[23] L. Clement, A. Hately, C. von Riegen, and T. Rogers, *UDDI Specification*, OASIS Technical Committee Draft Version 3.0.2, Oct. 2004. [Online]. Available: http://uddi.org/pubs/uddi_v3.htm

[24] K. Sakamura and N. Koshizuka, "The eTRON Wide-area Distributed-system Architecture for E-commerce," *IEEE MICRO*, vol. 21, no. 6, pp. 7–12, 2001.

[25] ISO, "ISO/IEC TR 9126-1:2001 – Software Engineering – Product Quality," 2001.