Scalable ID/Locator Resolution for the IoT

George Roussos
Department of Computer Science
Birkbeck College, University of London
London, UK
g.roussos@bbk.ac.uk

Paul Chartier
Praxis Consultants Ltd
Chippenham, UK
paul.chartier@praxisconsultants.co.uk

Abstract—**Identifier-locator separation is a mechanism that has been used successfully to support fine-grain object mobility and persistence. Support for legacy identification systems is seen as critical for the rapid adoption of the IoT. Thus, network-based identifier resolution and meta-data discovery services are already widely recognized as one of the core ingredients of the IoT architecture. This paper introduces a secure and scalable approach for the provision of such services derived as an application of the Handle System, which is extended with the components required for ID/Locator resolution on the IoT. This scheme is applicable to practically any IoT-based ID/Locator specification incorporating those established for different flavors of RFID and wireless sensor networks, and supersedes previous proposals such as the Object Naming and the OID Resolution Service. To highlight its advantages, we deploy our scheme to the long-standing artifact identification scheme specified under ISO/IEC 9834. In particular, we introduce a mapping for embedding OID Unique Item Identifiers in the Handle address space; several new HDL data-types defined so as to address common information provision needs in this context; and outline operational and use considerations. We conclude that the approach proposed here supports object mobility, fine-grain security compatible with international IoT requirements, especially with regards to governance, and multiple domains of control at the item level combined with superior scalability both with regard to identifier address space and system size.**

Keywords-component; ID/Locator system, identifier resolution, discovery service, Object Identifier.

I. INTRODUCTION

Internet Protocol addresses were introduced over 30 years ago and were designed for a relatively small global network of computers, where mobility was rare and participating nodes relatively homogenous. As a consequence, an early design decision established IP addresses as both a means to identify the end-point of communication and to specify its location within the network. The implicit specification of this dual role for IP addresses had unforeseen at the time consequences, and has recently led to severe problems in the core routing substrate of the Internet due to the limitations it imposes on the Border Gateway Protocol (BGP) caused by the growth of routing tables at the non-default route domain, and the limited support for multi-homing and mobility. To be sure, the dual role of IP addresses is today seen as a severe limitation on the Future Internet and the separation of identifier and locator is considered necessary, especially in the context of current IoT developments were object mobility is the norm rather than the exception and a variety of legacy and novel non-IP identifier schemes must be accommodated with a global operational framework and service provision.

Indeed, the use of non-IP identifier schemes is so widespread for a variety of material objects, locations and even digital artifacts that demanding a re-start from a clean slate would not be feasible either from a financial or an organizational point of view. Electronic Product Codes, Object Identifiers, Ubiquitous IDs and a variety of other schemes widely employed in RFID and barcode encodings, often in an industry-specific manner, are in current common use for the identification of billions of already tagged objects. Hence, a scheme that is capable to seamlessly support the integration of these artifacts in the IoT would represent a major breakthrough towards their incorporation into this system.

In this paper we introduce a novel extensible identifier-locator resolution scheme that places no restrictions on the choice of identifier or locator specification. We describe its main features and elements, and identify the main properties of the proposed system showing how they match requirements for its international operation. We highlight the adaptation of legacy codes for integration with the IoT by considering the popular case of Object Identifiers (OID) and by investigating the main issues related to the processing of such codes by a resolution and discovery service, paying particular emphasis on Unique Item Identifiers (UUIIs) commonly referred to as serials. Note that OIDs are standardized through ITU-T Rec. X.660 (2008) | ISO/IEC 9834 with UUIIs specifically the subject of section 9834-8:2009.

Our point of departure is the recent specification of scheme-specific services published by EPCglobal [9, 10] and ongoing work carried out by ISO/IEC JTC 1/SC 6/WG 9 and ITU-T Q12/SG17 on an OID resolver specification currently known as X.oid-res or the SG17 ORS (OID Resolution Service). In contrast to that work, we adopt an open and inclusive approach informed by the extensive experience and experimentation in this domain within the Internet Engineering Task Force (IETF) and specifically the work carried out on the HIP (RFC 5201) and LISP (RFC 6115) protocols and on scalable Internet-scale resolution systems [8]. Our goal is to identify a high-performance and scalable universal solution in the sense that it can support a wide variety of ID/Locator schemes integrating existing, under development and yet-to-be-developed schemes, and in the sense that it is based on an system architecture that lends itself to practical

implementations that provides high-performance with guarantees of scalability with respect to the number of serials and of geographic and administrative distribution. We also consider necessary that any such solution must provide security guarantees and in particular address the often-overlooked requirement for access control at the serial level.

The main finding of this paper is that it is indeed possible to develop an open, secure and scalable ID/Locator resolution service for the IoT incorporating item-level information and supporting diverse identifier schemes including legacy serials. We achieve this by extending the Handle System [11] from which our service inherits its security and scalability properties. The paper is organized as follows: In Section II, we review the main requirements for ID/Locator resolution on the IoT and associated discovery services and critically review the features offered by existing systems and current proposals. In Section III, we further motivate our work by relating it to the needs of OID with specific reference to UUIIs in Section IV. Section V, briefly discusses the Handle System and in Section VI we introduce our extensions and the implementation of the OID resolver. Section VII discusses additional HDL data-types that could further facilitate the discovery process and in Section VIII we consider the proposed approach from the point of view of alternative identifier systems. We conclude with a summary of the main findings and discuss directions of future work.

II. IDENTIFIER RESOLUTION ON THE IOT

ID/Locator resolution and associated discovery and information services are often seen as two of the main features of the IoT. In this context, such services are considered to be the means by which to alleviate the restrictions imposed by the resource constraints of IoT devices, which can only carry only limited quantities of object-related meta data and object histories. At a higher level of abstraction, identifier resolution is the process by which a code is mapped to its network location often represented as a URI and possibly linked to associated meta-data. Note that there is a fine line separating resolution from other types of query processing that are often seen as more extensive tasks providing additional features and guarantees such as consistency and isolation and state persistence. Contrary to this, resolution is seen as a lightweight process providing relatively simple mappings between different systems or providing the glue required to form integrated solution from distinct systems.

For example, within the Domain Name System (DNS) resolution refers to the mapping of a human-readable internet host or network names to their IP address equivalent for instance, linking the name www.aimuk.org to the IP address 81.21.75.152 and vice versa. Another example of resolution is the linking of a Document Object Identifier (DOI) such as 10.1007/978-1-84800-153-4 to a web location where relevant document-related meta-data can be retrieved (in this case a web page on Springerlink.com).

This concept was first employed within the EPCglobal system for the development of the so-called Object Naming Service [2] resolver that was designed to map product class level information retrieved from a Serial Global Trade Item Number (SGTIN) to an associated service point and service description. For instance, within this system the EPC code 075861.0434687.400 would be mapped to the following data:

```
EPC+EPCIS http://reference.verisignepctest.com
```

which indicate that further information about the implied product class (and possibly the serial) is available at the above location which can be queried using the EPC Information Service specification (essentially, an information repository specified in terms of capture and query APIs). ONS was developed on-top of the DNS in the sense that ONS records are maintained as so-called Name Authority Pointer records within the DNS and can be queried using standard DNS resolution tools available in virtually all internet-connected computers.

Note that the resolution process as described above does not imply the implicit definition of a one-to-one mapping between the two entities but simply that for each input query it will produce a related output (if such a mapping is defined for the particular data input). As a consequence, codes provided for resolution may be mapped to several outputs, and vice versa, multiple codes can be mapped to the same output as the examples below show. For example, every SGTIN code corresponding to the pattern 075861.0434687.* will map to the same service location as above.

Despite its simplicity, the above approach does not fully satisfy the requirements of the IoT and of course suffers from the well-documented security problems of DNS [8]. ONS is increasingly seen as a deprecated technology and EPCglobal has indicated a shift towards the EPC Discovery Service in the form currently developed within the EPCglobal Software Action [4, 5, 6] Group on Data Discovery. Data discovery is seen as a means to link together data repositories across the supply chain by providing a chain of individual pointers to independent EPC Information Service instances.

A protocol similar in scope and operation to the EPC Discovery Service is the so-called Extensible Supply-chain Discovery Service (ESDS) [15]. ESDS also provides persistent management of links to the sequence of custodians of particular serials and replicates records of significant events during their lifeline. Development of this specification appears to have been abandoned in 2009 in favor of the EPC specification.

One direct consequence of the loose definition of the resolution process is that in different contexts it can acquire widely different meanings. For example, the resolution of a DNS address is often a single-step process that involves a single request for a local (caching, non-authoritative) domain server. In some cases, the same request can lead to the cascading of communication messages, which can be either visible to the resolver that issued the query (for example via redirect messages) or invisible (since the local DNS server locates and queries the authoritative source of information). Note that the DNS specification allows for the return of non-authoritative responses that is, replies that do not consult the principle source of the mapping information, a necessary feature designed into the system as a way to reduce communication and delay. However, this functionality is also the most often abused feature of the system since it can be used to insert misleading information a problem often called cache

poisoning. The result of this is that even if the correct information if available, it may still be invisible to clients whose local resolution systems have been penetrated by an attacker.

Furthermore, the resolution process can involve multiple steps that extend beyond the boundary of a specific system or protocol. For example, the complete resolution of an EPC SGTIN code may first employ the ONS service to locate a repository where additional data related to the specific identifier are held, and then use a different protocol (for example, EPC Information Service Query API profiles over XML SOAP) to retrieve the specific data of interest (for example, color attributes related to the entity item of interest).

The resolution and discovery service model described above in the case of EPC has been the basis for the more recent work within ISO for the specification of an OID Resolution Service (ORS) carried out by SG17. This multi-step approach is closely followed by the draft SG17 ORS specification with the former step named public and the latter step named application-specific resolution. Similarly to the ONS, the ORS uses the same NAPTR DNS records to redirect to external full-query capable services.

The argument in favor of the use of the DNS as a resolution mechanism for the IoT is based on work on application layer mobility carried out within the scope of the Session Initiation Protocol (RFC 3261). SIP is a mechanism developed to mitigate the lack of connections at the transport layer for UDP applications, and its appeal transcends the Internet as it is also used within the 3GPP's Long Term Evolution (LTE) System Architecture Evolution to support Voice over IP (VoIP) applications (LTE is an all-IP network). With SIP, user identities are represented as URIs consisting of the personal identifier and the relevant SIP domain that has issued the particular identifier, for example:

```
sip:first.lastname@somedomain.net
```

In this way, sessions are bound to the SIP URI, not a specific network attachment point, which is obtained by querying the DNS for a specific type of record called Naming Authority Pointer (NAPTR) and passing the URI as parameter. The DNS resolves this query returning a prioritized list of network locations and associated access methods. These records can be updated in real-time applications, so that whenever users move, they update the URI binding with their new location, typically their new IP address, with the intention that communication remains unhindered.

While this *modus operandi* has a certain attraction as a mechanism for the support of ID/Locator resolution on the IoT, there is a strong argument that it does not match well the requirements for such a system. For example, the DNS naming hierarchy has a well established structured developed along organizational and national boundaries which does not accommodate the resolution of identifiers, as the identifier scheme has its own structure and hierarchy which does not easily map on the DNS. As a consequence, the ONS and the ORS mentioned earlier introduce new root top-level domains for identifier resolution, which operate in parallel to the DNS and thus does not represent as true integration of the two systems. Moreover, the above mechanism is designed with the view to support association of one URI to one or a small number of network locations and although it is possible to somewhat extend this, it does not fit well the requirement of IoT for possible multiple locators (for example, geographic, network, meta-data service and so forth) the inclusion of which quickly degrades the performance of the system. It also prevents the addition of commonly used meta-data (and related data types) to the identifier record, a feature that can significantly improve performance of applications.

Moreover, in the case of OID specifically there are additional limitations that restrict the applicability of the DNS. OIDs follow a tree-structure that defines richer relationships between codes for example parent, child and sibling, and in practice differentiates between terminal and non-terminal nodes that must be processed differently. This observation suggests that although there are benefits in re-using the DNS public infrastructure, the specific features, capabilities and mappings required for OID resolution are not fully satisfied. Notably

• Experience with DNS implementations implies that the resolution process does not scale well with the amount of data associated with a particular identifier and as a consequence DNS cannot accommodate persistence of meta-data beyond a simple mapping to a target URI.

• Typically, network administrators manage DNS services at the zone level and for this reason there is no provision within the system for actors outside this group to create or manage names. Specifically, there is no provision for a per-name administrative structure making the system unsuitable for general-purpose OID code administration.

• Performing sibling queries, which are considered a core feature of the OID resolution process, requires a so-called zone transfer (and further processing on the client side). However, zone transfers are considered a security violation and are disabled in most systems, so in practice this is not a viable alternative.

Finally note that France Telecom currently maintains non-UUID OID information at the public repository at oid-info.com which is however of unofficial and voluntary status. This is a closed system and has no provision for administrative delegation to constituent Registration Authorities and as such it is wholly inadequate for the role identified in this paper.

Overall, successful resolution and discovery services for OID codes are best characterized as a relatively rich resolution process. This implies that the simpler model supported by basic directories such as the DNS, are not adequate to fulfill all its requirements. At the same time, it is quite unnecessary to incur the significant overhead imposed by full-scale querying using the relational model for example, especially in the context of UUIDs. For this reason, we view the development of a resolution and discovery service as a mid-point between this extremes and indeed one that can be carried out effectively employing modern Internet resolution techniques.

III. MOTIVATING IOT APPLICATIONS

Object Identifiers originated in 1985 and are still specified through the work of ITU-T SG 17 and ISO/IEC JTC 1/SC 6 and their specification standardized ITU-T X.660 | ISO/IEC 9834 series, and as the name implies are one of several alternative object identification schemes. OIDs are defined as a hierarchical tree-like structure: each arc in this tree is numbered and objects are identified through the sequence of numbers representing the path from the root of the tree to the specific node representing the object in question. The standards also defines a hierarchy of registration authorities that are responsible for assigning arcs below their position within the tree and are free to delegate the task to a subordinate registration authority. For example, the OID identifier for the so-called mobile RFID mCode system used in Korea is defined in dot notation as 2.27.1, in ASN.1 notation as {joint-iso-itu-t(2) tag-based(27)} and as a Unicode-encoded International Resource Identifier (IRI) as oid:/Tag-Based/1.

OIDs can be used to identify a variety of artifacts beyond RFID and are commonly used for ASN.1 types and modules, X.550 and LDAP attributes, HL7 patient medical information and Simple Network Management Protocol Management Information Bases as well as a variety of other identification codes including bar codes. Over ninety five thousand class-level OIDs have been assigned since their introduction, making them clearly a highly successful system.

At its simplest, OID resolution may be defined as the direct mapping of an OID in numeric or dot notation to its corresponding IRI or ASN.1 representations, for example:

```
1.2.250.1 -> oid:/ISO/Member-Body/250/1
1.2.250.1 ->{iso(1) member-body(2)f(250) type-org(1)}
```

which is the approach adopted by the oid-info.com system, or resolved to a URI where can be further queried for meta-data related to the code, for example:

```
1.2.250.1  ->  XMLRPC+http://www.iso.org
```

which is the approach adopted by the SG17 ORS. However, both approaches appear to be too simplistic in the case of OID and do not fully capture the requirements of the resolution process for several reasons:

• A single tag may (often) carry more than one OID codes. For example, one OID may be used to identify the product class and the unique item identifier within this class, and a second to provide the product expiration date.

• Some OID schemes are time-specific in the sense that their interpretation changes depending on the time that the code is interpreted. For example, IATA-assigned OID codes are regularly re-used and as a consequence refer to different items of luggage and flight number over even relatively short periods of time and thus its interpretation is dynamic rather than static.

• It is sometimes the case that when an OID is retrieved what is actually required by the application is the complete or partial list of its children or siblings within the OID tree structure.

One particular situation of practical importance is when a query relates to the position of a particular code within the OID tree such in the case of terminal child nodes. The reason is that UUIIs are available only for this type of node and thus provide the only basis on which to recognize serials. This is particularly critical to differentiate between codes containing a UUII and those that do not, from the potentially multiple retrieved from a single tag.

Finally, in the following subsections we outline two particular application scenarios that are representative of the intended use of this system.

A. *Out of sequence actions*

The popularity of RFID baggage handling has increased significantly in recent years and systems of considerable size are already operational [1, 16]. Sorting using IATA-encoded tags in most cases is carried out locally that is under normal operating conditions decision making is at the reader controller level. But when an item of baggage cannot be routed using such local information, a condition that may often imply that the item may be lost, then the resolution process can reveal tracking information in a timely manner so that the particular item can be rerouted automatically. With an estimated 42 million items of baggage lost every year this facility can play a considerable role in reducing associated costs especially in the case of inter-airline and inter-airport communication.

B. *End-of-life product management*

One of the main drivers for Networked RFID has been the capture of richer information at various stages of a product life cycle that would allow more effective decision-making [5]. However, early experience seems to indicated that the cost-benefit analysis of this application is not yet favorable for several of the stakeholders involved and as a consequence, this approach has not found adequate support in the marketplace. With increasing pressure on manufacturing to take responsibility for end-of-life management of their products, the tradeoffs involved appear gradually to be changing and recent investigations appear to suggest that the availability of such product information can have positive impact on product recovery decisions with RFID thus providing the required capability for data collection, especially during the pre-sorting and grading stages of the end-of-life process [3].

IV. UNIQUE ITEM IDENTIFIERS

Although there is often discussion about a truly universal scheme of unique item identifiers, this is far from reality and a more realistic way to consider this is within domains or namespaces. Uniqueness is rarely persistent over many years, and a number of the identifier schemes permit codes to be re-used after a period of time that is relevant to the particular domain. At one extreme is the IATA Baggage Identification Number that can be re-used a number of times a year, but because its prime function is for tracking and tracing an item of baggage between the time it is checked in and collected by the passenger, this is understandable. In contrast, the ISBN scheme for books was designed so that serials are permanently assigned and indeed since 1968 when the scheme started, there have been no duplications (other than errors).

UUIIs typically follow a hierarchical structure that can be generalized in the following way:

```
Company or Organization
        Product (or Category) code
                Serial Number
```

Although UUIIs are often structured in this hierarchical format, in some cases the presentation of the code can include additional attribute data. In the following sub-section we consider unique item identifier schemes for different domains.

A. ISO/IEC 15459 Unique Identifiers

The scheme originated as a European Standard, EN1572, for a multi-industry transport label. The reason for this was to allow products from three primary European industries (automotive, chemical, electronics) to cross over between the industries and to use common transport facilities. Over the years, ISO/IEC 15459 has had additional parts added to it, effectively to provide an alternative to the GS1 code structure. ISO/IEC 15459 actually includes the GS1 code systems within the scope of each part of the standard. But since resolution and discovery is considered separately within the EPCglobal system, our primary focus here is on the industrial applications of ISO/IEC 15459. The unique identifiers are:

Part 1	Transport units
Part 4	Individual items
Part 5	Returnable transport items (RTIs)
Part 6	Product groupings

An organization such as Odette, representing the European automotive industry, is assigned an Issuing Agency Code (IAC). This is used as a prefix to a true domain code. In some cases, the code structure belongs exclusively to the particular domain using it (as in the case of Odette), but there are other codes such as "UN" for Dunn & Bradstreet business identifiers that have simply been adopted as a convenience. In fact, in the automotive industry, the codes for Odette, JIPDEC/CII, and Dunn & Bradstreet are used depending on regional variations and ownership of global organizations.

Despite the fact that several organizations have registered to use ISO/IEC 15459, it seems that only relatively few make complete use of these codes. In some cases, such as for Dunn & Bradstreet the intended use appears to be as a convenience to the users of the DUNS company identifier and similarly, for the case of the UPU.

Moreover, the extent that the IAC is actually used in the application layer is rather unclear. Certainly, it is used where there is an overlap between the domains registered under 15459 rules, and is encoded in data carriers. So there are two variants that apply depending on industry and/or company practice and requirements:

• Adding the IAC as a prefix to whatever hierarchical code follows.

• Using the IAC as a transition mechanism, but not holding it on the internal databases, and just using the basic code, sometimes even stripping that to its component parts.

B. Other Common UIIs Encoded in Data Carriers

In this sub-section, we look at some examples of other codes that differ significantly from the ISO/IEC 15459 traceability codes already discussed. For baggage handling, IATA assigns a 10-digit Baggage Identification Number that is used in a common way throughout the air transportation sector for identifying an item of baggage checked in for loading onto the plane. Although it is a sector-specific code, that sector has complete dependency on and usage of the code.

The International Blood Transfusion Service has a primary identifier schemes for individual donations of blood and other primary transfusion-related products. We have included this code structure because each of the fields carries a unique numeric identifier, but currently when encoded in a bar code it carries a unique identifier header.

The library community uses a unique identifier defined by the ownership domain of the library, which is generally acceptable for loan transactions with members as it effectively establishes a structured closed system. An increasing amount of material is shared between libraries under different ownership, requiring open system solutions. Within recent years, an international library code has been developed – the ISIL defined under ISO 15511, with its own Registration Authority. The concatenation of these two components (ISIL + library unique identifier) provides for a unique identification of every loan item in every library in the world. Already, over 10,000 libraries worldwide provided details of their catalogues so that 1.4 billion items are listed (see www.worldcat.org).

The US-based Air Transport Association has a shared responsibility with IATA, but has a particular focus on aircraft engineering standards. It uses as a base for its unique identifier a CAGE "company" identifier for the various products, followed either by a company-based product code and serial number or, in the case of some larger units, just the company and serial number. The CAGE codes are managed nominally on behalf of NATO, but are assigned with international scope – for example for the Australian military. The code is also applied to commercial aircraft components.

C. Other potential Item Codes Covered by ISO

Besides the highly relevant Registration Authorities for ISO/IEC 15459 and 15961, there are many ISO Registration Authorities for different types of items.

V. THE HANDLE SYSTEM

In our investigation of resolution and discovery service candidates we considered a variety of alternative approaches starting with a clean-slate approach but maintaining the advantages of reusing suitable systems and technologies wherever appropriate. After a critical appraisal of the possible solutions we opted for the Handle System [11, 12, 13], a general-purpose resolution system, since it provides a highly efficient infrastructure that incorporates best practice and can be readily tailored to the task of OID service provision.

Indeed, the Handle System matches well the requirements identified in the previous sections for OID resolution and discovery and moreover it offers features that can produce

significant efficiency gains compared to other alternatives. In particular, the Handle System supports:

- a two-level hierarchical service model,
- a scalable implementation,
- fine-grained authentication and access control, and
- an open and open-source implementation.

Handle is probably best known for providing the technical foundation for identifying digital content through the Digital Object Identifier (DOI) System. DOI is an application of Handle providing extended facilities specific to the particular application domain for example, policies regarding scope and application, business models and related application tools. The DOI System is managed and operated independently by the International DOI Foundation, a not-for-profit membership organization. Although DOI is the most prominent application of Handle, there are several others including DSpace, an open source repository application for delivering digital content to end-users, the Entertainment Identifier Registry (EIDR) for movie and television assets and the ORP service within the Global Environment for Network Innovations (GENI) a virtual laboratory for at-scale networking research.

Of particular significance in this context is the issue of scalability, as any OID resolution and discovery service would have to provide effective and efficient support for a potentially very-large number of serial and participating organizations. Distribution is a core ingredient of Handle that as a consequence it is capable of scaling well across two dimensions:

1. Number of requests. Within a site, servers can be replicated thus balancing the load associated with serving client requests thus maintaining low latency. If a site is required to server a very high number of requests then one or more secondary sites may be deployed through transparent replication.
2. Size of data. Individual namespaces can be spread across multiple servers within a particular site with the distribution of data guaranteed to be distributed evenly through the use of a purpose-specific hashing strategy.

Furthermore, the Handle System represents a mature technology as there is considerable practical experience in its implementation and operation at global scale. Indeed, according to the statistics published by CNRI, the Handle Registration authority, there are currently over 1,000 handle services located in 64 countries representing six continents. The number of registered prefixes currently exceeds 200,000. Handle services are operated by a large variety of user organization ranging in size and scope for example, user federations, national libraries and laboratories, and universities and research groups. Notable within them is the DOI implementation, which has over 45 million registered handles. The Global Handle Registry, which resolves first-level prefixes, receives on average 68 million requests per month with another 50 million resolution requests per month relayed through web-based proxy servers.

VI. OID RESOLUTION WITH HANDLE

In this section we outline the facilities of the recommended process for OID resolution based upon the infrastructure and facilities of the Handle system. The overall resolution process starts with an RFID reader that retrieves an OID code from a tag in its vicinity, which subsequently converts OID dot format and passes on to the OID Resolver for further processing.

Following the discussion of the following section, in this context it is appropriate to view OID Resolution as Handle application, similar in spirit to DOI as discussed in the previous section, but addressing the specific requirements of this domain. How such an application would operate is probably best illustrated via an example and we proceed by working out the details of one such example below.

First, we note that handle identifiers or simply handles have the following structure:

```
<Handle> ::= <Handle Prefix> "/"<Handle Suffix>
```

For example the handle

```
10.1045/april2006-paskin
```

has handle prefix 10 that indicated this to be a DOI. The derived prefix 10.1045 identifies that this is a DOI published by the D-Link magazine. The suffix april2006-paskin identifies the specific DOI which in this case resolves to an article published in the April 2006 issue of the D-Link magazine authored by Normal Paskin and entitled “Identifier Interoperability: A Report on Two Recent ISO Activities.” The assignment of further derived prefixes can be independently managed by the DOI Foundation and does not require the involvement of the Handle system. D-Link Magazine can create further derived prefixes under 10.1045 as well as suffixes as they see fit. Furthermore, the specific format and the rules applied to the interpretation of the suffix are almost entirely up to the owner of the derived prefix namely the D-Link Magazine and can have any semantics.

A handle is submitted for resolution to the Handle system by a local software agent often referred to as the local resolution agent. The Handle Foundation currently offers and supports a variety of implementations of such resolution agents implemented in different programming languages. In response, the handle system will return a typed set of results (and it is possible to define new result types beyond what is currently available under the Handle specifications). For example, possible results can include URLs and locations.

The Handle System supports an infrastructure of globally accessible servers that can receive and process requests from clients and return result sets. The first step in the resolution process is to identify the specific server, called the Local Handle Server, that is responsible for a specific prefix and relay the request to it. The LHS will further process the request possibly relaying to further LHSs responsible for specific derived prefixes and eventually return an authoritative result.

Let us now turn our attention to the resolution of OID specifically taking the case of IANA assigned Baggage Identification Codes as example. We can assume that the RFID reader has captured data from a tag with root-OID

1.0.15961.12, relative-OID 1 and compacted data 1234567890, which has subsequently concatenated in full dot format as

```
1.0.15961.12.1.1234567890
```

The task at hand is the transformation of the above OID representing a specific item of luggage into a form that can be passed directly to the handle resolver so that related information can be retrieved.

The first step in this process would be to add the OID-specific Handle prefix. We have registered a prefix for this task with CNRI, which operate the registration authority on behalf of the Handle Foundation. The OID prefix used during our experimentation is set to 10673 and we have configured our system to use this. Using this prefix, we introduce a straightforward way to structure handles used for OID, that is by creating derived prefixes for the complete OID arc, with the UII becoming the handle suffix. Applying this reasoning to the example above the resulting handle obtained is

```
10673/1.1.0.15961.12.1.1234567890
```

This is now a fully compliant handle and can be passed on to any handle resolver for processing. The result returned by the resolution process is appropriate in that it is specific to the particular derived prefix and each provider can specify its specific form using existing or new handle data types. For example, the resolver can specify within its query the specific type of result requested for example data type 10320/loc (a reserved data type specified by the Handle System administrator) specifies that the request should return an XML document of one or more locations associated with the specific OID, for example the resolution of the above item of luggage may indicate its location at the following coordinates

```
<locations>
   <location id="0"
   href="http://maps.google.co.uk/maps?q=51.522394,+-
   0.130881" country="gb"
   weight="0" />
</locations>
```

This allows for considerable flexibility in both the administration of the scheme and the interpretation of serials. One the former issue, several alternative approaches can be developed for the implementation of the scheme that allow for example the delegation of the administration of the 54321.1 scheme to ISO but its actual operation to the Handle Foundation or alternatively both tasks can be handled by ISO. Further, it is possible to make the Handle Foundation responsible for the operation of the top level arcs for the whole of OID for example up to and including standard root OID definitions.

Regarding the interpretation of serials, it is possible to extend the operation of a local handle server with application specific requirements that would implement specific rules based on regular expressions on the suffix. In the example above, the first two digits of the UUII 1234567890 could be given specific meaning for example they could represent a particular carrier responsible for assigning this code.

VII. Extended Datatypes

As noted in the previous section, a handle query specifies the particular data types that it relates to (with the possibility to query for all). Indeed, every handle record consists of one or more typed values of the form

```
HDL:Type:Value
```

and Handle clients rely on the type information to determine the correct action for a particular value returned as the result of a query. The Handle System does not validate any (type, value) pair internally and this task is explicitly reserved for applications, which can choose suitable behaviors. Common predefined data-types include administrative information for example contact email for queries, and associated meta data for example a generic data element and URLs which can be used to easily replicate the ONS and ORS capabilities for instance. In the case of URL values, the default behavior of the resolution client is to simply select the first URL value in the list of values returned by the handle system. Because does not require a specific ordering of this list, there is no intelligent selection of a URL to which the client should be redirected. The 10320/loc handle value type used above was developed to improve the selection of specific resource URLs and to add features to the handle-to-URL resolution process. Note that within Handle the prefix 10320 has been set aside for use only by the Handle System administrator with the intention of registering handle application types.

Data type 10320/loc specifies an XML-formatted handle value that contains a list of locations. Each location has a set of associated attributes that help determine if or when that location is used. The overall list of locations can include hints for how the resolving client should select a location, including an ordered set of selection methods. Resolvers can apply each known selection method, in order, to choose a location based on the resolver's context (the HTTP request in the case of the proxy server) and the attributes of each location.

The attributes for the set of locations, as well as each location entry in the set, are open-ended to allow for future capabilities to be added in a backwards-compatible way. A small number of attributes have been defined as "standard" that all resolvers should understand. Value types can thus be defined locally within a specific sub-domain and for example can be associated with particular prefixes only. Indeed, the administrative authority responsible for the creation of particular handles is free to define and employ any handle value types that are required within the intended domain of application.

However, this flexibility may lead to problems for applications that may be unable to comprehend the semantics of returned values. A solution to this problem is in current development within the Handle System whereby registration authorities can register new handle value types (HVT) on the system using specific prefixes. This feature is intended to replace the current convention that had the basic data types registered under the special handle 0.Type. New types can be defined to be complete handles and are registered within a novel Handle System service the so-called Handle Value Type Registry (HVT-R). This registry provides a variety of advanced features such as search capability for types. Note that new

types can reference other HVTs or non-handle value types such as common encodings for example MIME.

TABLE I. SUGGESTED OID-RELATED DATA TYPES

Name	Type	Value
Current location	URI	Current network attachment point and method of access
Last URI	URI	Last known location
Geographic Location	URI	Geographic location
Meta-data repository location	URI	Location of repository holding streaming and real-time data, meta-data, tracking and other information
Creation date	Date	Date of OID serial created
Expiry date	Date	Expiration of OID serial
Provenance checker	URI	Location of provenance validator (e.g. original manufacturer)
History trail	URI List	Complete sequence of custodian URIs

Table I summarizes some of the OID-specific data types we have proposed for use by this system. Data are encoded using the Protocol Buffers specification, an extensible high-performance serialization system for structured. In addition to the performance consideration, the use of Protocol Buffers for the encoding of OID data types within Handle can facilitate cross platform client applications since data can be exchanged transparently across language implementations.

VIII. SUPPORT FOR OTHER IOT IDENTIFERS

The entire purpose of using OID structures for RFID item management was to align with ISO/IEC 9834 and RFC3061 A URN Namespace of Object Identifiers, which was published in February 2001. All of this came about based on the original approach for the data protocol to be completely dependent upon ASN.1. The decision to base everything on object identifiers has proved to be sound, because it enables new application domains to adopt RFID and make use of the established infrastructure. In essence, this is a simple split mapping presentation directly to physical artifacts, illustrated in the Table II below.

A. Monomorphic-UUII Supporting Legacy Structures

As mention in Section IV, one approach for using the Monomorphic-UII is to support the encoding of the UII in a manner that, when decoded, emulates the encoding in bar code. In this case a prefix is added. Although this can be either the Application Identifier of GS1/EPCglobal, the decision of that organization is to encode all UUIIs in RFID tags using EPC codes. So the only option is for the IAC to be identified by the Data Identifier (DI). Using an ISO/IEC 15459 example with the IAC assigned to Odette, the UUII is prefixed with the DI, shown below in bold:

25SODA1B2A1B2C3D4E5

A compliant ISO/IEC 15962 interrogator / decoder will use the proposed machine-readable AFI table, which will also include the associated DIs, and create the appropriate root-OID. In this example, the OID structure for resolving is the same as that encoded using the no-directory encoding and is therefore:

```
1.0.15459.4
```

TABLE II. DATA TO PHYSICAL ARTIFACT MAPPINGS

	Presentation	Device & System Implementation
Upper Level	Using the OID structure + UII	Application interface of 15962 From a "smart" interrogator supporting the device interface standards In the ISO/IEC 24791 RFID System Management Protocol Potentially with the Internet of Things
Lower Level	Encoding to ISO/IEC 15962	In RFID memory Across the air interface From a basic interrogator to data protocol processor

B. Monomorphic-UUII Using URN Code 40

This is illustrated in Section IV using the ATA example, which could be registered in early 2010. The concatenated OID + UII, creates a new OID, which can be resolved to the lowest level of the UII hierarchy. Because the dot separators are retained in the decoded UII, the output from the Interrogator / 15962 decoder automatically creates the complete OID ready for resolving. Only the application administrators can define the encoding scheme, so this needs to be documented when applications are made to register the data constructs.

C. Monomorphic-UUII Using URN Code 40 for ISO/IEC 15459

The basic structure described in Section IV has already been accepted as a generic solution for RFID (i.e. with the registration of AFIs and Root-OIDs). As discussed in 5.3, the present approach outlines a set of solutions that is narrowly focused on backward compatibility.

Here we put forward an outline proposal that is based on a Monomorphic-UII, and will also make use the URN Code 40 structure. The code structure would be as follows:

```
1.0.15459. {part}.{IAC}.{next}.{more layers…}.{last}
```

For illustration, assume that Odette decided to adopt this approach. The example cited in 5.3 would be encoded as:

```
1.0.15459. 4.OD.A1B2.A1B2C3D4E5
```

It would be the responsibility of the company encoding the data to ensure that the dot separators were placed correctly. On decoding, a company that wanted the legacy structure would have a standardized procedure to strip out the dot separators. Those that wanted to use the resolver capability would us the code as is.

The structure proposed above, of the 15459 Part number being in the UUII, reduces this to be an request to the ISO/IEC 15961-2 Registration Authority for a single AFI that is of benefit for any 15459 IAC, including those yet to be assigned.

An alternative, which produces a slightly shorted encoding of the UII, is for the application to the RA to cover three AFIs, one for each 15459 part, shifting the part number component from the UII to the root-OID. This might not be acceptable, as it would consume a significant proportion of the AFI code stack.

IX. Discussion and Conclusions

Scalable ID/Locator resolution and discovery based on universally unique item number identifiers is a core ingredient of the IoT. In this paper we have shown how to develop a universal resolution service as an application of the Handle System and have investigated in detail the specific case of Object Identifiers incorporating item-level information. We have further explored the costs and benefits of this service and discover that it provides a good fit to this task. Firstly, by inheriting the fine-grain security of Handle, our proposal is capable to provide similarly fine-grain access control of item-level data. Second, it supports a highly scalable architecture, which can accommodate both an extended identifier address space and a large number of client requests. Third, in can incorporate a variety of identifier schemes including legacy (such as OID, EPC, Virtual Entity Identifiers and a variety of related ISO specifications) and new as yet unspecified identifiers tailor made for the IoT within one single system and a structured address space. Fourth, it supports international and organizational requirements for the administration of identifier schemes. And last but not least, there is an open and robust software implementation that has been used for operational deployment with great sucess.

Our recent experience with the proposed ID/Locator system has offered encouraging evidence that this can be a widely applicable system. One specific question that remains to be considered is the integration of IoT technologies within the existing (and indeed the future) Internet, and in concluding this section we offer a few comments related to current IETF and IRTF activities on architecture. Notably, there is currently an active discussion on future routing architecture(s) the current status of which is summarized by the recently issued RFC 6115. Within this discussion, the issue of ID/Locator separation is prominent although it does not consider the IoT scenarios, which are the focus of our attention. Indeed, work seems to be primarily concerned with the case where participating nodes have higher communications and computing capabilities akin to at least 6lowpan wireless sensor network nodes, and a full or partial implementation of an IP software stack. This point of view sets considerable challenges to the development of IoT and appears to exclude the billions of objects that are already automatically identifiable and possible candidates for membership to the IoT.

Nevertheless, there are significant similarities between the work presented here and several of the ID/Locator protocols currently in discussion. Notably, proposals for a Compact Routing in a Locator Identifier Mapping System (CRM) and Global Locator, Local Locator, and Identifier Split (GLI-Split) can be accommodated by our proposals. Although clearly the artifacts employing the identification schemes considered in this paper would not support the computational requirements of the above protocols, it is still possible within an edge system to provide surrogating services, which is in line with the current approach of IETF specifications for resource constrained devices. Therefore, we believe that our current findings justify further exploration of the costs and benefits of the proposed approach and we are currently pursuing this avenue of investigation.

Acknowledgments

The authors would also like to thank Larry Lannom and Norman Paskin for extensive discussions on the Handle System. This work is supported by CASAGRAS2 under research contract IST- 258440 with the European Commission.

References

[1] P. Chartier, An Overview of ISO RFID Standards Applied to IATA Baggage Handling, IATA Technical Report, 2005.

[2] EPCglobal, Object Naming Service (ONS) Version 1.0 EPCglobal Ratified Specification, 2006.

[3] M. Harrison, D. McFarlane, A.K. Parlikad, Y.W. Chien, Information management in the product lifecycle - the role of networked RFID, INDIN04, Berlin, 2004.

[4] K. Kailing, A. Cheung, S. Schönauer, Theseos: A Query Engine for Traceability across Sovereign, Distributed RFID Databases, Proc. ICDE, Istanbul, Turkey, 2007.

[5] D. Kiritsis, A. Bufardi and P. Xirouchakis, Research issues on product lifecycle management and information tracking using smart embedded systems, Advanced Engineering Informatics, 17(4), pp. 189-202, 2006.

[6] C. Kürschner, C. Condea, O. Kasten and F. Thiesse, Discovery Service Design in the EPCglobal Network, Lecture Notes in Computer Science, 4952, 19-34, 2008.

[7] R. Rantzau, K. Kailing, S. Beier, T. Grandison, Discovery Services Enabling RFID Traceability in EPCglobal Networks, Proc. COMAD, Delhi, India, 2006.

[8] V. Ramasubramanian, and E.G, Sirer, The design and implementation of a next generation name service for the Internet. Proceedings ACM SIGCOMM '04. ACM Press, 331–342, 2004.

[9] G. Roussos, Networked RFID: Systems, Software and Services, Springer SMB, London, 2008.

[10] S. Sarma, D. Brock and D. Engels, Radio frequency identification and the electronic product code, IEEE Mirco, 21(6), 50 – 54, 2001.

[11] S. Sun, L. Lannom, B. Boesch, Handle System Overview, Internet Engineering Task Force, RFC 3650, 2003.

[12] S. Sun, S. Reilly, L. Lannom, J. Petrone, Handle System Protocol (ver 2.1) Specification. Internet Engineering Task Force RFC 3652, 2003.

[13] S. Sun, S. Reilly, L. Lannom, Handle System Namespace and Service Definition. Internet Engineering Task Force RFC 3651, 2003.

[14] F. Thompson, Extensible Supply-chain Discovery Service Schema, IETF Internet Draft, thompson-esds-schema-04, 2009.

[15] M. Young, Extensible Supply-chain Discovery Service Concepts, IETF Internet Draft, draft-young-esds-concepts, 2004.

[16] T. Zhang, Traceable air baggage handling system based on RFID tags in the airport, Journal of Theoretical and Applied Electronic Commerce Research, 3(1), 106-115, 2008.