Efficient information lookup for the Internet of Things

G.F. Marias, N. Fotiou, G.C. Polyzos
Mobile Multimedia Laboratory,
Athens University of Economics and Business
47A Evelpidon, Athens, 113 62 Greece,
{*marias,fotiou,polyzos*}*@aueb.gr*

***Abstract*—The Internet of Things is an emerging paradigm that allows the association of information with objects. The information about an object is stored in databases, distributed around the globe, maintained by various stakeholders that participate in the object's supply chain. A Discovery Service (DS) is responsible for collecting all these database URIs and feeding them to clients that query about an object. DSs are usually centralized and are designed to effectively aggregate as many information sources as possible, rather than trying to respond optimally to unforeseen user queries.**

In this paper we propose a novel, Information-Centric Networking (ICN) inspired, architecture that eliminates the need for a separate DS, enabling at the same time multi-ownership and flexible management of information that is associated with an object. In our ICN approach, information about an object is organized in *scopes*. Each scope has its own access control rules allowing easy control of information dissemination. Moreover scopes can be hierarchically organized, creating complex access structures that can reflect business relationships. In our architecture companies provide information about an object to the appropriate scopes, but they never lose control of this information. To access information associated with an object,a user queries the scope that corresponds to the desired context, which will forward the query to a service that will respond with the appropriate data. The query will not reveal any extra information, not authorized to be retrieved.

***Keywords*-Information scoping, Information-Centric Networking, service discovery.**

I. Introduction

In recent years and mainly due to the advances of Radio Frequency Identification (RFID) technology, the Internet of Things (IoT) is emerging as a promising paradigm which allows the association of an object with information. It is expected that vast amounts of data, distributed in databases all over the world will be available for each object. An obvious problem that emerges is how can this data be effectively mapped to an object, i.e., how can requests about the information associated with an object be answered in a fast and reliable way. A solution proposed to this problem is the creation of a Discovery Service (DS). In its basic form a DS is a centralized service that collects references to information associated with an object and when requested by a user it responds back by sending all of them. It is obvious that such an approach will face scalability and security problems. In this paper we try to solve these problems, based on two observations; a) users usually do not care about all the available information about an object, but they rather want to access specific data which is closely related to their context (e.g., a customer of a store is interested in learning the prices of the specific store and not of every store) and b) if information is structured in a way that reflects business relationships, it can be disseminated more effectively–as it should be easier to locate it–and more securely–as it would be easier to define access control policies on it.

In order to effectively handle information lookup we propose an Information-Centric Networking (ICN) architecture in which information items are organized in *scopes*. Scopes are a hierarchical structure that imitates existing business relationships and allows the mapping of an object identity to specific data. Since a scope can be directly mapped to an entity or to a relationship, it can be easily decided what information will be accessible through a specific scope. Similarly a user can easily decide which scope should he use in order to access the desired data. The proposed architecture defines operations that allow information advertisement, access and manipulation. The function requirements of a DS as identified by Evdokimov et al.[1] are also satisfied, i.e., multiple stakeholders can publish information about an object and information can be updated or deleted by those stakeholders. Moreover our architecture is independent of the identification scheme used and it allows the definition of authorization procedures. Finally our architecture is context-specific, i.e., the response to a user query is closely bounded to his context (e.g., location, identity, or access network)

The operation of the proposed architecture can be summarized in the following: stakeholders of objects advertise objects attributes to the appropriate scope, users, that are interested in learning the value of an object's specific attribute query the corresponding scope, if the user is eligible to access this attribute, the stakeholder is notified and the value of the queried attribute is send to the user. Even from this brief description it can be observed that information is structured, therefore it can be searched in a fast and scalable way. Moreover the proposes architecture is based on strong security foundations as scopes allow access control policies definition and object's stakeholders have full control over their attributes and their values.

The structure of the remainder of this paper is as follows. In Section 2 we discuss related work in this area. In Section 3

978-1-4673-1239-4/12/$31.00

we present our architecture. Finally Section 4 presents our conclusions as well as plans for future work.

II. Related Work

The Object Name Service (ONS) [2] is a process, standardized by EPCglobal, that enables the location of services associated with a given Electronic Product Code (EPC), using the DNS. The EPC binary string is translated into a Uniform Resource Identifier (URI), which is then converted to a Uniform Reference Name (URN), according to [3]. This URN is compatible with the existing DNS infrastructure therefore DNS can be used in order to locate a service endpoint. As it can be understood, there is an 1-to-1 map between the EPC and the service endpoint, therefore this solution has many limitations. Moreover ONS's architecture implies that a single organization, or a country, will be the central point for providing authorized global discovery services, introducing several concerns on how the different national and international privacy restrictions and regulations can be fulfilled.

Our architecture does not suffer from these limitations. Information scopes can be separately administered and their operation does not depend on any centralized entity. Moreover an object identity can be mapped to multiple information items

EPCglobal has also formulated a DS, but its details are still unpublished [4]. Nevertheless, various articles–such as [1], [5]–give us a high level overview of this architecture. EPCglobal's DS is a centralized repository in which service endpoints are advertised. A client wishing to receive information about an EPC issues an appropriate query to the DS and receives back all service endpoints that can answer that question.

Our approach is more distributed and there is not a single point in the network in which all information is accumulated. Moreover, in our architecture, when a client issues a query does not receive any unnecessary information, he only receives the answer tho his query.

The Bridge project has also investigated the potential of a lookup service [6]. The project created four variants of a centralized architecture; a directory of resources that is similar to the EPCglobal DS, a service that notifies resources about new queries, a service that notifies users that have made queries about new attributes, and a query propagation architecture. In all cases, both users and DS learn more than the necessary information.

In our architecture scopes become aware only of certain number of attributes and (eligible) users learn only what they ask.

SOCRADES [7] uses a centralized application service catalog which users can ask for services using queries.

In our architecture information is organized in scopes, which enables simpler queries.

Various other research efforts–such as [8], [9], [10], [11]–investigate the possibility of using a P2P discovery service. Papers [8], [9], [10] use a DHT where service endpoints are advertised, whereas [11] proposes a Multipolar ONS which still uses DNS as ONS, for compatibility, but decentralizes the ONS root to avoid unilateral control of it. In all cases a client will receive all endpoints as a response to a query.

III. The proposed architecture

The basic entities of the proposed architecture are: the *object creators* and the *object holders* (hereafter will be referred to as creators and holders). Each object has one creator and multiple potential holders. In terms of a supply chain, the creator of an object is its manufacturer, whereas the holders are the various resellers, or customers. Creators and holders create *information structures* which then access using the *operations* of the architecture.

A. Information structures

Each object is identified by a unique ID assigned by its creator and it is associated with multiple <attribute,value> pairs created by its creator as well its by the holders.

For each object we consider the following classes of <attribute, value> pairs:

1) Global Read-Only
   These are attributes well known to the users. The values of these attributes are set by creators. This class includes attributes that remain unchanged throughout the lifetime of an object, e.g., the ingredients of a drug. Moreover the values of global attributes of two identical objects are the same (e.g., all "X" drugs have the same ingredients).
2) Global Editable
   These are well known attributes for which each holder can set its own values. This class includes attributes which are important in the supply chain of an object–therefore all parts of the chain should refer to them in the same way–but their value can be modified during its lifetime. Moreover the values of the global editable attributes of two identical objects are not necessarily the same. Price is an example of global editable attribute (the price of drug "X" can vary from pharmacy to pharmacy).
3) Local
   These are attributes known only to individual holders. This class includes attributes that help holders to better manage the objects they possess. An example of local attribute can be the number of the shelf in which a product is stored.

In order to clarify the above concepts we consider as an example the supply chain of a pharamaceutical (drug). The company that creates a drug (therefore its creator) assigns to it a unique identifier that is stored in a tag attached to its packaging. Moreover it creates the following attributes:

description, ingredients, weight and *price*. From the above attributes description, ingredients and weight are global read-only and they are set by the company, whereas price is a global editable attribute for which each pharmacy (holder) can set its own value. Moreover a pharmacy that sells this drug may set its pharmacy-specific values that can help it, for example, to manage its stock (such local attributes can be, for instance, the number of stored items and the number of sales). Each pharmacy may as well agree with other pharmacies on the usage of common attributes in their business-to-business services.

Each entity (creators and holders) maintains a lookup service that maps attributes to values. These services are accessed through *scopes*. A scope can be considered as a meta-service which–in its simplest form–accepts as input attribute queries and forwards them to the appropriate lookup service. A scope is controlled by a number of entities and it defines the boundaries within which values for global editable attributes are applicable as well as the visibility of a local attribute (i.e., to which extend this attribute is reachable). Each scope is accessed through a URI, which is also used as the *Scope Identifier* (SId). The form of SIds depends on the implementation details, it can be for example a URL or a flat identifier. Moreover an SId controls as well the visibility of a scope, for example if IP addresses are used as identifiers, a scope identified by a private IP will be accessible only in the local network of the scope owner whereas a scope identified by a global IP will be accessible from any part of the Internet. Scopes reflect the various business relationships and they can be hierarchically grouped. To better understand the notion of scopes we extend our drug supply chain example. Suppose that there is a pharmacy with 4 branches in different cities. This pharmacy may have a scope for each branch and a parent scope that represents the pharmacy as a whole. Each branch specific scope may provide information about the available stock in each branch whereas the parent scope may provide statistics for the total sales of a drug.

In scope hierarchy an approach akin to inheritance in programming languages is followed: child scopes can handle requests for attributes on behalf of their parent scopes. In addition to the entities-specific scopes, a unique global (root) scope per object is also considered. The global scope provides access to the global read-only attributes and every other scope inherits it. An entity may participate in multiple scopes, for example a pharmacy may have a scope of its own, it may participate in the scope of an e-commerce web site and it may participate in a scope with other pharmacies with which it shares its stock. A scope specific implementation is transparent to the users; a scope can be implemented using a single database, a distributed hash table or a hierarchical scheme. In any case every scope ultimately maintains a forwarding table that forwards attribute queries to the appropriate lookup services.

Each entity in the system has a unique identifier as well as multiple *roles*. As an example a user with identity C, can have the role of customer of pharmacy A and employee of pharmacy B. Similarly to attributes, identities and roles may belong to a global or to a limited scope.

Figure 1 illustrates our drug supply chain example. The drug ID–as assigned by Company A, which is its creator–is MedX. Company A has set two global read only attributes, namely Name and Ingredients, which are advertised in the global scope. As scope identifier (SId) we are considering the IP address of a scope. The global scope can be accessed from anywhere as it uses a global IP address. As it can been seen the global scope maintains a forwarding table that maps attributes to a lookup service URL. In this example we are considering two holders; the pharmacy PhA and its branch, branch1. Each holder has created its own scope using inheritance, that reflects the existing business relationships: pharmacy PhA is a customer of company A and branch1 is controlled by PhA. The created sub-scopes are assigned a private IP address therefore they can be accessed only from the Internal networks of Pharmacy A and branch1 respectively. With this a network layer mechanism limits the dissemination of each value. Of course higher level access control policies can be also considered. Each sub-scope has a forwarding table which contains a link to the lookup service for the attributes defined by the holders as well as a link to the parent scope for the inherited attributes.

B. Operations

Our architecture defines the following operations; *Scope Creation*, *Attribute Advertisement*, *Attribute Subscription* and *Value Forwarding*. With the *Scope Creation* operation, a global or an entity-specific scope is created. In the later case a placeholder for the newly created scope in the scope hierarchy should be defined. The new scope learns all the entries of the forwarding tables of its parent scopes. The forwarding tables of a scope are populated by the creators and the holders using the *Attribute Advertisement* operation. Each entity can set access control policies to the attributes it specifies. These access control policies map identities and/or roles to the actions they can perform on each attribute. Access control policies may also contain additional constrains such as time and location restrictions. Access control policies are implemented in the scope and can take advantage of scope inheritance for the definition of finer grained rules, i.e., access control rules can be more generic in scopes that are higher in the hierarchy and more specific in lower level scopes. Entities can un-create scopes and un-advertise attributes.

A user requests the value of an attribute by invoking the *Attribute Subscription* operation. With this operation a user asks a scope for the value of a specific attribute. A special class of attribute subscription messages is also considered, that enables a user to learn all the attributes of a scope.

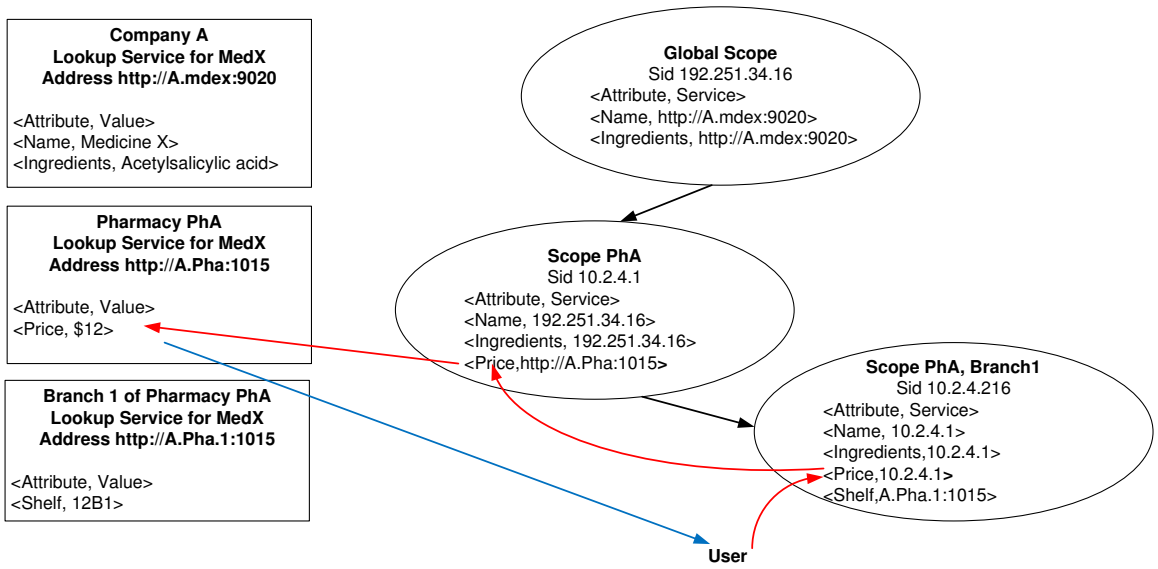

Figure 1. An example of information structure . Black arrows denote inheritance. Red arrows show a "subscribe value" message for the attribute "price" sent by a user to scope PhA, branch1.

Each attribute subscription message may include information about user context and identity. This information is used for applying access control policies as well as for deciding which value should be returned as an answer to a user's query (e.g., the attribute "price" may have different value for regular customers than for normal customers). Each subscription message is evaluated by the scope and in case of success will result in a value being forwarded from the lookup service to the user. The latter is achieved using the *Value Forwarding* operation.

In each *Attribute Subscription* an object ID, an attribute name and the scope from which the attribute should be obtained, have to be specified. As an example, a doctor wishing to learn the ingredients of drug A, should query A's global scope for the value of the attribute "ingredients", whereas a customer of the pharmacy B wanting to learn the price of drug A should query the scope of pharmacy B for the value of the attribute "price" of A. SIds of global scopes are considered to be well known, e.g., they can derived from the object identity. On the other hand the SIds of entities-specific scopes can be learned by out-of-band mechanisms. Such mechanism may include the usage of human readable SIds which will be provided as input to the RFID reader by the users and the development of DNS-like mechanisms for accessing them or the usage of "default scopes" within a network (e.g., all requests about a drug originated in a pharmacy's local network, can be automatically directed to the pharmacy' scope). Default scopes can be learned using DHCP-like mechanisms or they can have a well-known

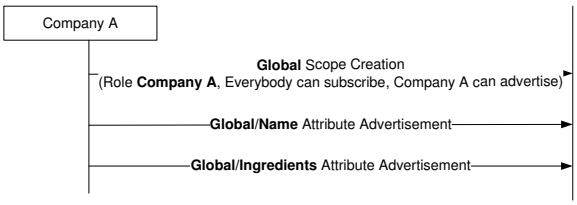

Figure 2. Sequence of operations, executed by Company A in order to create the example information structure

(local) identifier (e.g., local IP address). A user request is initially validated by the scope in order to determine its accordance with the access control policies and if it is valid it is forwarded to the appropriate lookup service or to the parent scope. Eventually the lookup service will respond to the user with the appropriate value.

In the example of Figure 1 we consider a user, located inside branch 1 of pharmacy Pha, who requests the value of the attribute "price" of MedX. User request is handled by the scope 10.2.4.216, which is the scope of branch 1. We assume that the user is authorized to make this request, so 10.2.4.216 scope uses its forwarding table and forwards the query to its parent scope which in its turn forwards it to http://A.Pha:1015, which is the URL of the appropriate lookup service. The lookup service finally forwards back to the user the value of the requested attribute.

Figure 2 illustrates the sequence of operations, executed by Company A in order to create the information structure depicted in Figure 1. As it can be seen Company A initially

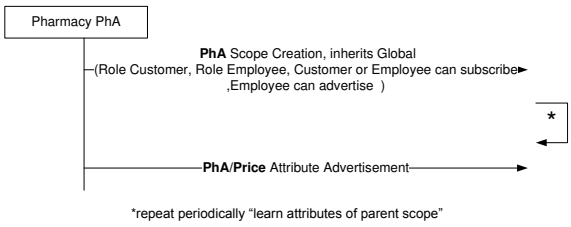

Figure 3. Sequence of operations, executed by Pharmacy PhA in order to create the example information structure

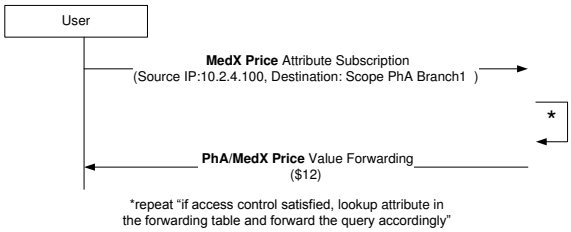

Figure 4. Sequence of operations execution by User in order to learn the value of the price attribute

creates a global scope, creates a new role in this scope, named *Company A* and assigns an access control policy: everybody can subscribe to attributes but only Company A can advertise attributes. Then, Company A advertises *name* and *ingredients* attributes under global scope. Similarly Figure 3 illustrates the respective operations, executed by Pharmacy PhA. It can be observed that since the created scope inherits the global scope, the attributes of the global scope should be referenced by this new scope. Moreover the global scope should be periodically examined for any attribute changes. PhA also advertises the *price* attribute in that scope.

Figure 4 shows the steps required by a user in order to learn the value of the attribute "price". Initially the user sends an attribute subscription message to the "default scope" that he has learned using a DHCP-like mechanism. The message contains the object identity, the name of the attribute, and user's IP address. User's IP address is used as a "proof" that the user is inside branch 1 therefore, he is a customer. The query is evaluated by the first scope and if the access control policy is satisfied the query is forwarded to the location defined in the forwarding table, which in that case is the parent scope. In the next step the query is forwarded to the URL of the lookup service. Eventually the value is send to the user.

C. Security

Each operation configures or triggers a security mechanism. With the scope creation operation the scope owner defines the roles that exist in this scope as well as an access control policy for that scope. The scope specific access control policy specifies who can advertise attributes and who can subscribe to attributes advertised in that particular scope. More fine grained publication policies can be set with the attribute advertisement operation. With the attribute subscription operation a user should also provide all the necessary credentials associated with the attribute as well as with the scope in which the query is sent.

Since each scope is separately administrated, the particular mechanisms used in order to satisfy the security requirements are scope implementation specific. In general each scope should provide the means to identify users and authenticate their messages, to enable users to express their access control policies in a descriptive manner, as well as to implement these access control policies. In our example users in the global scope can be identified using public keys provided by a third trusted party (so the role "Company A" should be mapped to a public key, and so on) and each message can be authenticated using digital signatures. Finally access control policy definition and implementation can be done using OASIS XACML [12]. In the scope owned by PhA different mechanisms can be used in order to identify users. Customers can be identified by their local IP address (if they are using PhA local network then they can be identified as PhA's customers) whereas employees can be identified using public keys issued by PhA.

An important feature of our architecture is that attributes values are only known to the lookup service and they are never revealed to the scoping mechanisms, this has the benefits that value owners can easily and with no cost modify the values they have specified for an attribute. Moreover they can verify that a scope decision was correct and refuse to sent a value in case of error. Finally attribute revocation can be effectively achieved: even if a scope is slow in deleting a revocated attribute from its lookup table, the lookup service can answer with an error if the value of a deprecated attribute is requested.

IV. Prototype

A prototype system based on the proposed architecture is planned. The realization is based on the information centric Publish-Subscribe Internet (PSI) architecture originally created by the FP7 project PSIRP[1] and now being further worked on and extended by the FP7 project PURSUIT.[2] PSI provides all the necessary functionality and structures required in order to implement the envisioned IoT architecture: PSI enables publication, subscription, and forwarding of information items and incorporates scoping and lookup mechanisms. [13] The approach is based on an extension of Blackadder [14]: an open-source node implementation following the PSI architecture.[3]

[1]http://www.psirp.org/
[2]http://www.fp7-pursuit.eu/
[3]https://github.com/fp7-pursuit/blackadder

V. Conclusion and Future Work

In this paper we proposed a novel architecture for information lookup in the Internet of Things. We argued that existing approaches that rely on discovery services can be improved. We introduced a new scheme that uses information scoping and does not require a separate service discovery mechanism. In our proposal, information about an object is advertised in a hierarchical structure of scopes, which enables fast and accurate information lookup. The scope structure reflects existing business relationships, therefore, it fits better the product supply chain. Moreover, an object's stakeholders never lose control of the information they posses: they can easily modify it, revoke it and protect it against faulty operation of the system; thus our scheme has strong security foundations. Since each scope is administered by a separate entity, their internal implementations can be different, therefore, the requirements of various stakeholders can be easily satisfied.

The work in this paper set the conceptual foundations of our architecture. Our future work is heading towards three main directions: protocol specification, system implementation, and development of security mechanisms. We intend to formalize the operations defined in our architecture, specifying a flexible messaging framework, capturing the requirements of the emerging applications of the Internet of Things. Our ultimate target is to create a flexible, efficient, and secure architecture. The scoping mechanism is expected to facilitate the development and deployment of access control solutions, which will protect sensitive data without violating business relationships.

VI. Acknowledgments

The work reported in this paper was supported by FP7 ICT project PURSUIT, under contract ICT-2010-257217.

References

[1] S. Evdokimov, B. Fabian, S. Kunz, and N. Schoenemann, "Comparison of discovery service architectures for the internet of things," in *Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), 2010 IEEE International Conference on*. IEEE, 2010, pp. 237–244.

[2] EPCglobal, "Epcglobal object name service (ons) 1.0.1, ratified standard specification with approved, fixed errata," 2008, http://www.gs1.org/gsmp/kc/epcglobal/ons/ons_1_0_1-standard-20080529.pdf.

[3] R. Moats, "URN Syntax," 1997. [Online]. Available: http://www.ietf.org/rfc/rfc2141.txt

[4] EPCglobal, "Epcglobal standards overview," February 2012, http://www.gs1.org/gsmp/kc/epcglobal.

[5] C. Kürschner, C. Condea, O. Kasten, and F. Thiesse, "Discovery service design in the epcglobal network," in *The Internet of Things*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2008, vol. 4952, pp. 19–34.

[6] Bridge Project, "Bridge WP02 high level design discovery services," 2007, http://www.bridge-project.eu/data/File/BRIDGE WP02 High level design Discovery Services.pdf.

[7] P. Spiess, S. Karnouskos, D. Guinard, D. Savio, O. Baecker, L. Souza, and V. Trifa, "Soa-based integration of the internet of things in enterprise services," in *Web Services, 2009. ICWS 2009. IEEE International Conference on*. IEEE, 2009, pp. 968–975.

[8] B. Fabian and O. Gunther, "Distributed ONS and its Impact on Privacy," in *Communications, 2007. ICC'07. IEEE International Conference on*. IEEE, 2007, pp. 1223–1228.

[9] N. Schoenemann, K. Fischbach, and D. Schoder, "P2p architecture for ubiquitous supply chain systems," in *17th European Conference on Information Systems (ECIS'09)*, Verona, Italy, 2009.

[10] B. Fabian, "Implementing secure p2p-ons," in *Communications, 2009. ICC'09. IEEE International Conference on*. IEEE, 2009, pp. 1–5.

[11] S. Evdokimov, B. Fabian, and O. Gnther, "Multipolarity for the object naming service," in *The Internet of Things*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2008, vol. 4952, pp. 1–18.

[12] R. Hayton, "Oasis: An open architecture for secure interworking services," University of Cambridge Computer Laboratory, New Museums Site, Tech. Rep., 1996.

[13] N. Fotiou, D. Trossen, and G. Polyzos, "Illustrating a publish-subscribe internet architecture," *Telecommunication Systems*, pp. 1–13, 2010.

[14] Kjllman, J., ed., "PURSUIT deliverable 3.2, first lifecycle prototype implementation (d3.2)," September 2011, http://www.fp7-pursuit.eu/.