

IDN-Travel: touristic data production and reuse beyond walled gardens

Riccardo Billero, Maria Chiara Pettenati, Lucia Ciofi, Franco Pirri

Electronics and Telecommunications Department, University of Florence, Florence, Italy

E-mail: {riccardo.billero, mariachiara.pettenati, lucia.ciofi, franco.pirri} @unifi.it

Abstract: The diffusion of social networks has boosted the user-generated content sharing in different application domains. At present travel-oriented social networks do not allow the users to organize their information with an easy reuse of data found on a Web-scale. In this paper we address the problem of Web of data reuse through the IDN-Travel Application, a solution that allows users to manage granular data that can be structured as desired, by reusing other information that may be everywhere in the Web. The present application is built on top of the InterDataNet architecture allowing users to collaborate on structured documents aggregating distributed data from the Web. In InterDataNet each data unit has an owner responsible entitled of its management. Each authorized user will (re)use data directly from its origin without duplicating it. This work aims to demonstrate the technical viability of this solution, allowing the user to (re)use data without be confined within the walled gardens established by social media.

The InterDataNet infrastructure solution handling auxiliary information related to data and documents reuse can therefore serve other application scenarios.

Keywords: Web of data, travel, data reuse, collaboration, content-centric architecture

1 INTRODUCTION

The widespread of social networks such as Facebook [1] and thematic social networks, such as LinkedIn [2], Flickr [3] and Slideshare [4] revealed with unprecedented relevance the Internet users interest in creating and sharing multimedia content [5], [6].

The tourism sector is one of those areas in which sharing of user-generated content is of broad interest; these information can be used as a way for the users: to keep digital memory of their travels, to share their experience with others, as a support for the organization of a new journey, using the experiences and feedback provided by others, etc..

This importance is demonstrated by the success of services such as sharing pictures of trips to social media like Facebook [1], or more oriented to the photographic

sector, like Flickr [3], up to a whole range of services specifically focused on travels like Trip Advisor [7], Tripntale [8], Triplogged [9], minube [10].

A major characteristic of these services is that they are "walled gardens" [11]: they manage the resources in a context dependent way and they do not allow other users and/or applications to reuse the same data, i.e. they provide limited interoperability capabilities.

At best, these services allow the use of mashups [12], such as showing maps provided by Google [13], embedding videos coming from YouTube [14], or using the Facebook "I like" button [1]. However, to make these techniques working, there should be an agreement made in advance between the parties, or at least the use of API specifications imposed by the included service provider.

Moreover, another of the characteristics of these services is that they ensure features that are desirable for the user, such as privacy and permission management, only within the specific application domain. In addition, the security management is entrusted entirely to the service provider.

Finally, these services are characterized by the fact that they focus on the application and not on the data. This fact contrasts with the direction towards a Future Internet [15] to shift the focus from applications to a content-centric approach.

There is a need to allow the user to manage resources that are easily sharable, reusable, beyond *walled gardens*, giving him the possibility to affect to the content a set of properties such as privacy, permission management, licensing, provenance, persistency, versioning.

There is a need for strategies to increase collaboration and then reuse, focusing on an approach concerning the data itself and not at an application-specific level, in order to decouple the information artifact from the application and make it independent.

The solution we propose in order to address collaboration at a content-centric level, is to avail of the services of the InterDataNet (IDN) infrastructural architecture [15] on top of which we develop IDN Compliant Applications.

This solution allows the user to exit in an "open prairie", instead of staying inside one of the walled garden available, while paving the way for the development of applications possibly reusing the same data.

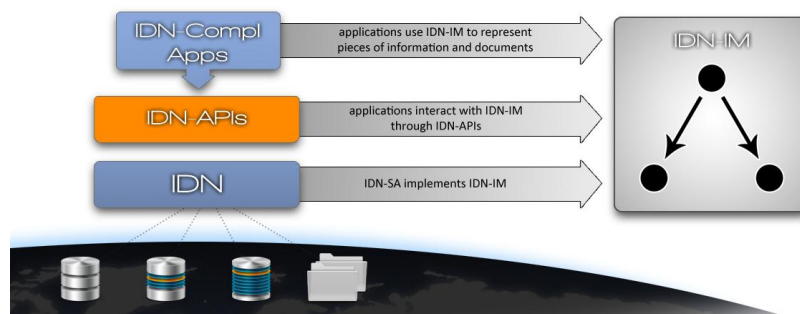


Figure 1: The InterDataNet architectural components and their mutual relationship: IDN-IM, IDN-SA and IDN-Applications

To demonstrate the viability of this solution in the context of tourism and travel, we present the IDN-Travel application case study and prototype. The IDN-Travel application gets its name from the fact of being an IDN Compliant Application, which is an application that meets the specifications of the IDN infrastructure, described in Section 3.

2 IDN-TRAVEL APPLICATION

IDN-Travel is an application for helping users to store in the cloud and share with other users their personal travel information, such as trips, monuments' and places' descriptions, diaries, photos etc.

In this Section we'll explain the characteristics of this application, giving a short description of some of the use cases.

2.1 An application for travellers

Let's consider a group of friends sharing a common passion for travels. After returning from a trip, friends gather to talk about the experiences and see taken photos.

They'd like to share the information concerning their travel, such as diaries, photo and so on, using the Internet.

They would also like to integrate what they have collected and produced with other information already available on the Web, such as monuments descriptions, other pictures, travel diaries and reuse them as part of the information relating to their trip tale.

2.2 Use cases

IDN-Travel allows the user to manage his own digital memories relating to travels; the user can view a list of trips where he has taken part or that have been shared with him by other users.

Selecting a trip, the user can display a wide range of information about it, as the period in which it was done, the participants, the sharers of the document relating to the trip, the destinations, a number of comments related to the trip, the travel report and a gallery of the best photos taken during the trip. Each resource can be clicked, thus opening a page related to it.

Hereafter we provide a short description of some of the use cases provided by the IDN-Travel application.

2.2.1 Creating a new travel

By pressing the "Create new journey" button, the user is able to create a document concerning a new trip. The application shows a page in which the user should insert a range of data relevant for the journey like the destination, the participants, the report and so on.

2.2.2 Viewing an existing travel

By the "My Travels" page the user has a comprehensive view of the travel he saved. Selecting a trip, the user access to a page concerning that trip, where some of the features of the trip are shown, as the destination, the participants, the report, the photo gallery.

2.2.3 Viewing photos

From any page that displays details about the trip, the user has an access point to the photo gallery. By selecting a particular photo of the gallery, the user can view it with a better resolution and see the information linked to it, with the comments left by other users.

3 IDN ARCHITECTURE

IDN-Travel is realized as an IDN Compliant Application, i.e. is an application made according to specifications of the IDN Framework (see Figure 1).

IDN Compliant Applications use a specific information model, named IDN Information Model (IDN-IM), to represent pieces of information and documents; these applications interacts with IDN-IM through IDN-APIs; finally, IDN-IM is implemented by the IDN Service Architecture.

In this Section we describe the IDN Framework, as worked out in previous work [15], focusing on particular on the IDN Information Model.

3.1 The IDN Framework

IDN Framework is described through the ensemble of concepts, models and technologies pertaining to the following two views (see Figure 1):

- IDN-IM (InterDataNet Information Model). It is the shared information model representing a generic document model which is independent from specific contexts and technologies. It defines the requirements, desirable properties, principles and structure of the document to be managed by IDN.
- IDN-SA (InterDataNet Service Architecture). It is the architectural layered model handling IDN-IM documents (it manages the IDN-IM concrete instances allowing the users to “act” on pieces of information and documents). The IDN-SA implements the reference functionalities defining subsystems, protocols and interfaces for IDN Document collaborative management. The IDN-SA exposes a REST [16], [17] IDN-API (Application Programming Interface) on top of which IDN Compliant Applications can be developed.

3.1.1 The InterDataNet Information Model

The IDN Information Model is the graph-based data model (see Figure 2) to describe interlinked data representing a generic document model in IDN and it is the starting point from which the design of IDN Architecture has been derived.

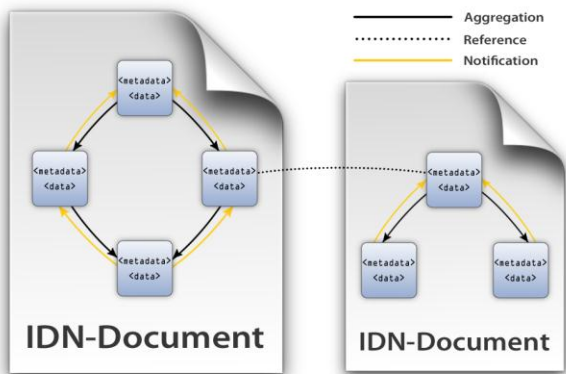


Figure 2: InterDataNet Information-Model

Generic information modeled using IDN-IM is formalized as an aggregation of data units, each of which is assigned at least with a global identifier and contains generic data and metadata. It is designed to have a minimal yet extensible set of properties like the definition of a responsible, an author, type of copyright and more. The abstract data structure is named IDN Node and it is the “content-item” handled by the “content-centric” IDN Service Architecture. The degree of atomicity of the IDN Nodes is related to the most elementary information fragment whose management is needed in a given application. It is also the basic exchange unit between different applications. The information fragment to be handled in IDN-IM applications is aggregated with other IDN Nodes to form compliant documents, i.e. a rich hierarchically structured content. The container is called IDN Document.

Both the IDN Node and the IDN Document are addressable by an HTTP URI and their contents (name,

type, value and metadata) are under the responsibility of a given entity. The node aggregated into an IDN-IM document is used (read, wrote, updated, replicated, etc.) under the condition applied/specified on it by its responsible. An IDN-Document structures data unit is composed of nodes related to each other through directed “links”. Three main link types are defined in the Information Model:

- aggregation links, to express relations among nodes inside an IDN Document;
- reference links, to express relations between distinct IDN Documents;
- back links, to enable the mechanism of notification of IDN Nodes updates to parent-nodes (back propagation).

The different types of links envisaged in IDN-IM are conceived to enable collaboration extending the traditional concept of hyperlink. Indeed the concept of link expressed by the “href” attribute in HTML tags inherently incorporate different “meanings” of the link: either inclusion in a given document, such as it is the case of the image tag, or reference to external document, such as it is the case of the <a> anchor tag. The explicit separation of these differences allows making the meaning explicit and consequently enabling the different actions, e.g. assigning specific authorizations on the resources involved in the link. In Figure 2 aggregation, reference and back links are illustrated.

3.1.2 The InterDataNet Service Architecture

IDN Service Architecture (IDN-SA) (see Figure 3) is made up of four layers: Storage Interface, Replica Management, Information History and Virtual Resource. Each layer interacts only with its upper and lower level but also relies on the services offered by IDN naming system. IDN Nodes are the information that the layers exchange in their communications. The IDN Node HTTP-URI allows interacting with the resource in CRUD style, using the four primitives GET, POST, PUT and DELETE.

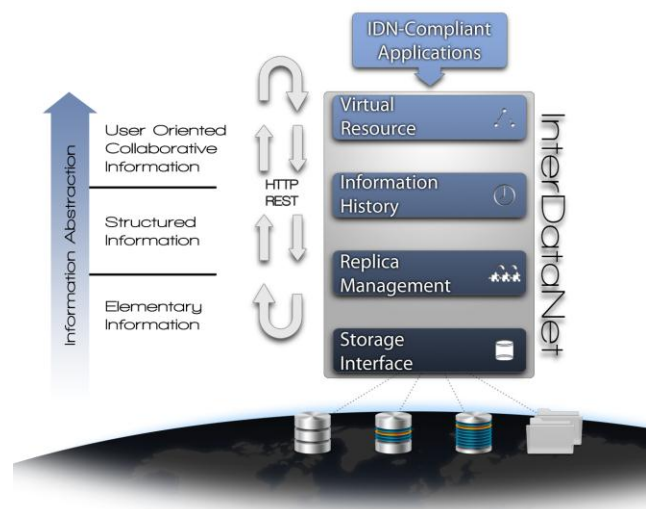


Figure 3 : InterDataNet Service Architecture

The IDN Compliant Application layer stands on top of the four layers of the Service Architecture.

The communications between IDN-SA layers follows the REST paradigm through the exchange of common HTTP messages containing a generic IDN Node in the message body and IDN Node identifier in the message header.

4 DESIGN OF THE IDN-TRAVEL APPLICATION

In this section we describe the approach for designing the innovative IDN-Travel application, first focusing on an IDN Document and then on the REST design of the application.

4.1 The IDN-Travel Document

The IDN infrastructure uses resources according to the REST architecture specifications terms [16]. In particular, these resources are the IDN Node and the IDN Document.

The key point of the REST approach is first to define all the resources; this is done in a declarative form and not in an algorithmic one (i.e., the REST approach is not about services).

The functions, where they exist, must stay in the business logic. The creation of resources is both identify the basic elements (i.e. the nodes) and decide the properties of the nodes and organize their relationships within an IDN document.

IDN adds to simple raw data a set of auxiliary information called metadata such as author, responsible, etc... The basic information are called IDN Nodes, because they are linked together in a Direct Acyclic Graph structure that we named IDN Document (see Figure 2).

The IDN Document creates complex information through the aggregation and correlation of Nodes, and the addition of a number of properties related to the collection itself, such as the author of this collection and so on.

The IDN Node is a resource that could be defined universal and it is thanks to the use of the IDN Node that it is possible to leave the walled garden; on the other hand the IDN Document may also be partly inside a walled garden and partly not.

The capabilities provided by the specific IDN Node are determined directly by the owner of that node; e.g. an IDN Node may have a read-only policy for anyone, and read and write access for authorized users.

At the time when the node becomes part of an IDN Document, the management policy of the node within the document is determined by the document itself, taking into account of any constraints imposed by the policy of the node.

One of the key features of the IDN-Travel application is the fact that the data itself establishes its visualization policy inside of the application and therefore the presence or absence of functions and interface tools used to edit data, delete it, etc...

For example, it is possible to define inside the IDN-Travel application the author of any node and the features that can be published; it is possible to accept comments from any type of user, whether registered or not; or it is possible to make the report editable only by registered users.

4.2 REST Design of the IDN-Travel Application

IDN-Travel manages each trip as an IDN Document. When a user creates a trip, IDN-Travel provides a special tree in which to store the subordinate resources, which are the resources that are associated with them to form a direct acyclic graph.

That is, each new travel document responds to a specific document template, defined by the application itself.

The template defines a set of minimum requirements to be met in order the document to be properly managed by the considered application. In particular the template has a specific structure for the provision of subordinate resources, adopted in the design phase of implementation of the application. For example, each trip consists of a route through one or more cities and for each city, there may be one or more visited sites, with a series of photographs relating to them. Moreover, each trip owns a report carried on the route.

In practice, the template is knowledge of the application layer that uses it as a way to define new IDN documents; the template is not part of the IDN Architecture, but uses the features provided by it. IDN Architecture comes into play when the user creates a new document (which respects the structure of this template).

The document model defined by IDN-Travel also includes a report for each trip, which is unique and will be implemented with an appropriate limitation of feasible operations; the report can be edited by all participants in the trip.

We assume that the <http://www.idntravel.com/travels/> resource is the "delegated resource" to the creation of new trips. The execution of an HTTP PUT request on a subordinated resource of it corresponds to the creation of a new journey, whose name is determined by the user.

Now, let's consider a user who desires to create a document concerning his last trip to Palermo. The process of creating a new document concerning a trip, transparent to the user, corresponds by the point of view of IDN-Travel to the performance of an HTTP PUT operation on the <http://www.idntravel.com/travels/> resource, cited above. Within this document "root" the user will gather all the experience on his trip.

The user can change the different parts of the model, such as the report, or he can create new resources as subordinated to those preset in the template; for example, the user can share his multimedia information, such as photos. The adding of a new picture corresponds to the performance of an HTTP POST method on a specific resource, related to that particular visited place.

It is appropriate to observe that the types of links defined in Section 3 allow IDN Architecture to automatically mashup the external data either it is an IDN resource or a simple HTML object such as images and other.

Using the reference link, the user can also add some reference to various resources distributed in the web whether they are documents IDN or generic resources.

For example, the user can add two reference links towards external resources, to describe the Cathedral of Palermo, to the Wikipedia page [18], or the official website [19].

The management of concurrent access in the phase of editing is resolved by using the HTTP ETag header. Each time a GET method is executed on a resource, the server responds by including an ETag header in the response, a unique identifier assigned by a web server to a specific version of a resource. The ETag allows to perform an HTTP conditioned PUT method on his value [20], so that the application can check if other users have made changes in the time interval between the GET operation, the user made, and PUT one, he'd like to make.

Since the documents created by the application IDN-Travel were stored according to the specifications of the IDN infrastructure, the reuse of data is facilitated. In fact, since each part of the document relating to the trip is saved in the form of IDN Node, any user who wants to create a new journey may use the information stored in it, subject to permission.

Not only that, but every single IDN Node can be safely reused within any other application that meets those requirements, thus providing the ability to optimize the reuse of data.



Figure 4 : The IDN-Travel application prototype

5 IDN-TRAVEL PROTOTYPE AND ISSUES TO BE ADDRESSED

At present, the prototype of IDN-Travel so far achieved has been implemented as a Web application, using HTML, CSS, JavaScript and jQuery.

The user interface is realized through the composition of objects that, for each block, have a view on information, so each page can be thought as a set of areas, each of which displays information relating to a particular resource (see Figure 4).

There are some aspects that have yet to be explored; first of all, the nature of the IDN-Travel application makes it suitable for use on mobile devices such as smartphones or PDAs. We are also directing our efforts towards the realization of this application (see Figure 5).

Another issue to be addressed is security and privacy. Even if the travel context that we are addressing is such that the issue of privacy is not a priority, because we are tapping into an entertainment context and used data are not strictly personal, since the used approach aim at enabling the information reuse, it is interesting to possibly ensure the management of the security and privacy needs.

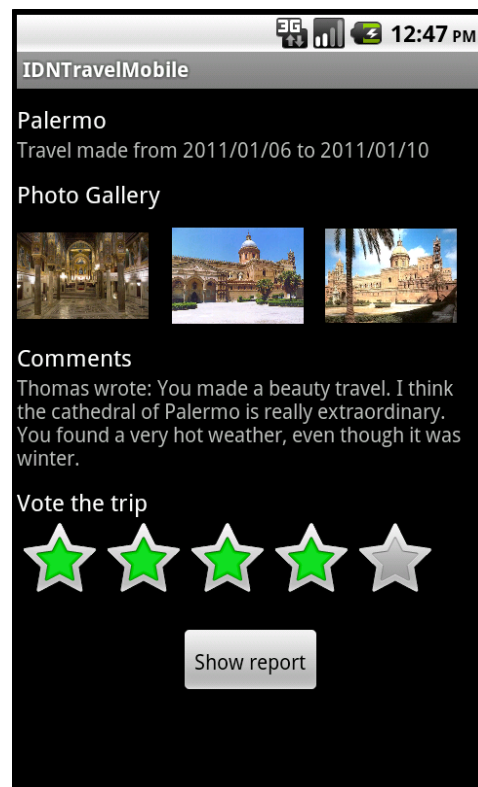


Figure 5 : The IDN-Travel mobile application

In other words the infrastructure is in charge of managing everything so that it works; thus becomes possible to develop IDN Compliant Applications that are able of managing sensitive data in particular contexts, as eGovernment or eHealth.

Finally, the use of other entities presupposes the existence of other applications for reusability; our proposal is to design applications by structuring their documents in "IDN style" or compatible.

The IDN infrastructure it is indeed able to manage simple HTML objects, such as tags, plain text, etc., addressable with traditional URLs and show them directly within the IDN Compliant Application

We are currently investigating the possibility of managing legacy objects such as MySQL databases and standard formats, seen as "chunks" of HTML or ATOM.

6 CONCLUSIONS & DISCUSSIONS

In this paper, we proposed the IDN-Travel application, in order to help users to manage their digital memories concerning trips.

By using the IDN infrastructure becomes possible for the user to exit in an "open prairie", instead of staying inside one of the walled garden social media.

IDN-Travel application moves the focus from an application-oriented approach to a content-centric one, toward the direction of a Future Internet.

Each resource has some specific characteristics that help to set the mode in which it is itself displayed.

Through the use of an IDN Document, each item has particular attributes, for example, the relationship between those who are looking at a resource and those who actually created it. By designing the document with the IDN infrastructure, these attributes are handled directly in the infrastructure.

The user interface of IDN-Travel can be considered a social media, because it allows the user to interact directly on the resource that is being displayed.

Obviously, the user can modify the resource if he has the appropriate permissions, for example because he is the creator of the resource or because he has the rights to edit it.

The management of the resource is entirely delegated to the IDN infrastructure, both for classical CRUD operations and for the management of privacy, permission management, security, and so on.

The traditional services described above, usually can manage mashup of data whose they know the type; by the use of IDN architecture, the IDN-Travel application can show all the resources that can be retrieved with an HTTP GET method: if the resource was also stored as an IDN Document, it can also be managed and changed.

We introduce an example to explain how to reuse the resources between different applications. Let us consider the existence of an application called IDN-Monument, adopted by the Italian Ministry of Cultural Heritage. This application is responsible for keeping track of the heritage within the Italian State. The application is implemented using the IDN infrastructure and, for each monument, is

defined a particular IDN Document that contains a variety of information relating to it as a collection of photos of the document, the description, the conservation status, the opening times for visitors, the cost of tickets, and so on.

The user is able to define as he wishes how the part of the document concerning a monument should be structured. He can also decide to import only some of the information defined by the application IDN-Monument; for example, he may decide to include just a photo and the official description of the monument.

The IDN infrastructure does not use an algorithmic API, but a declarative one and is easier to be integrated. Indeed it has been demonstrated that standardizing a document schema is easier than standardizing an API and its behaviour [21].

InterDataNet is an open project of the University of Florence [15].

Acknowledgements

We would like to acknowledge Mr Luca Capannesi for the technical support in the implementation of IDN Service Architecture prototype.

References

- [1] <http://www.facebook.com>
- [2] <http://www.linkedin.com>
- [3] <http://www.flickr.com>
- [4] <http://www.slideshare.net>
- [5] Wilson, C., Boe, B., Sala, A., Puttaswamy, K.P.N. and Zhao, B.Y., *User interactions in social networks and their implications*, Proceedings of the 4th ACM European conference on Computer systems, pp. 205-218, 2009
- [6] Benevenuto, F., Rodrigues, T., Cha, M. and Almeida, V., *Characterizing user behavior in online social networks*, Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference, pp. 49-62, 2009
- [7] <http://www.tripadvisor.com>
- [8] <http://www.tripntale.com>
- [9] <http://www.triplogged.com>
- [10] <http://www.minube.com>
- [11] Berners Lee, T., *Long Live the Web: A Call for Continued Open Standards and Neutrality*, <http://www.scientificamerican.com/article.cfm?id=long-live-the-web>
- [12] IEEE Internet Computing, vol.12, no 5, September/October 2008
- [13] <http://maps.google.com>
- [14] <http://www.youtube.com>
- [15] Pettenati, M.C., Ciofi, L., Pirri, F. and Giuli, D., *Towards a RESTful Architecture for Managing a Global Distributed Interlinked Data/Content/Information-Space*, 2011 edition of the Future Internet Assembly book, *Future Internet: Achievements and Promising Technology*, Springer. Available in open access, 2011
- [16] Pautasso, C. and Wilde, E., *RESTful Web Services: Principles, Patterns, Emerging Technologies*, Tutorial at WWW2010 (Raleigh, North Carolina), April 27, 2010.
- [17] Richardson, L. and Ruby, S., *RESTful Web Services*, O'Reilly Media, Inc., Sebastopol, CA, USA, 2007.
- [18] http://en.wikipedia.org/wiki/Palermo_Cathedral
- [19] <http://www.cattedrale.palermo.it/>
- [20] Nielsen, H. F. and LaLiberte D., *Editing the web: Detecting the lost update problem using unreserved checkout*, W3CNote, 1999
- [21] Weis, T. and Wacker A., *Federating Websites with the Google Wave Protocol*, IEEE Internet Computing, vol. 15, no.3, May/June 2011, pp. 51-58.