

InterDataNet: A Scalable Middleware Infrastructure for Smart Data Integration

Franco Pirri, Maria Chiara Pettenati, Samuele Innocenti, Davide Chini, and Lucia Ciofi

1 Introduction

The Internet of Things (IoT) is a vision in which the Internet extends into our everyday lives through a wireless network of uniquely identifiable “objects” or “things” [22]. While the “Internet of Things” perspective has been focused on enabling the networking capabilities of the Things, the “Web of Things” perspective proposes to integrate the networked “Things” into the Web [20], thus making them available as resources, i.e., documents. The advantage of this integration approach is that the “Things” can be treated like any other Web resources, leveraging on the REST architectural style to provide low entry barrier. This would eventually enable the loosely coupled “Things” to be reused in different contexts and applications. In the meantime, the Web itself would be evolving [9] and undergoing significant transition reflected by the massive effort to make the WWW evolve from a file-server to a database paradigm, with the purpose in mind of creating an “interlinked data Web” doing for data what the World Wide Web did for documents; this vision is known as the Linked Data approach [1, 2]. In the words of the creator of the Web itself, “The Linked Data approach is a Web of Things, conceptually, because it gives the thing a notion of its own identity in the Web. Tying an actual thing down to a part of the Web is the last link – the last mile” [13]. Taken together the two approaches, the Internet/Web-of-Things and the Web of Data concur in the vision of a data-web (lower case is deliberate) evolving to become part of a world-wide database to be both exploited and enriched by internetworking “things” on a Web-wide scale. In order for such view to come true, the realization of “ubiquitous network environment,” that is the integration of heterogeneous sensing devices, heterogeneous actuators, and Web of Data is required. At the state of the art few convincing architectural styles for such integration exist, while the academic and industry interest in this domain is growing [16, 19]. Some projects, for instance [22], though proposing innovative applications, focus on IoT applications developed as

F. Pirri (✉), M.C. Pettenati, S. Innocenti, D. Chini, and L. Ciofi
Electronics and Telecommunications Department, University of Florence, Italy
e-mail: franco.pirri@unifi.it; mariachiara.pettenati@unifi.it; samuele.innocenti@unifi.it;
davide.chini@gmail.com; lucia.ciofi@unifi.it

dedicated systems with specific environments in mind with low extensibility and reuse capabilities. Other works, aiming at offering objects integration architecture on a Web-wide scale [3, 8, 11, 20] though targeting the scalability as a main driving principle, focus on making all resources available via standard Web mechanisms, dealing with “things” much likely as other Web resources that are documents. InterDataNet approach brings scalable interoperability to IoT leveraging on the Web of Data paradigm. Such an approach entangles using flexible identification mechanism allowing rich description, management, and interoperability of heterogeneous objects and things. A key step in this direction is making data “smarter.” Indeed, the smarter are the data describing the objects, the easier is sharing, linking, and processing between distributed applications. Making data smarter and using such data at an infrastructural level, means moving smarts into the data itself rather than hard coding them into software applications and/or smart objects.

In this chapter, we provide an overview of the InterDataNet architecture designed to enable heterogeneous objects networks to expose and integrate their smart data. At the core of the system sits the InterDataNet middleware that defines an object Information Model and the related Service Architecture operating on it in order to provide: (a) *a scalable and open service to support a consistent reuse of objects identifiers*, that is a global reference and addressing mechanism for locating and retrieving objects in a Web-wide scale; (b) *a set of transparent application-services functions*, namely historic data management and replica management.

This chapter focuses on discussion of architectural elements rather than implementation details. For more complete technical and implementation details, see [10] and forthcoming papers.

2 Grounding Principles and Design Paradigms of the IDN Framework

To fulfill the main driving design criteria [6, 11, 18–20] of extensibility, low entry barrier, interoperability, and scalability of the architecture we propose, we adopt the following set of conceptual and technological design paradigms:

- *The system has to be layered*; layering [23] is the architectural pattern to pursue scalability and legacy data integration at infrastructural level. This is achieved, implementing the functionalities at different IDN layers as network devices.
- *The design of middleware following Service Oriented Architecture (SOA) approach* [12, 15] to allow the development of loosely coupled and interoperable infrastructural services that can be combined into more complex systems.
- *The system has to adopt a REST style (Representational State Transfer) paradigm* to make InterDataNet an explicit resource-centric infrastructure [7, 21]. REST architectural style that emphasizes scalability of component interactions, generality of interfaces, independent deployment of components, and intermediary components to reduce interaction latency, enforce security, and encapsulates

legacy systems. WWW is one instance of a real distributed system based on REST. WWW enjoyed the scalability and growth as a result of a several design principles defined in REST. In REST approach, every resource (sources of specific information) is uniquely addressable, each of which can be referred to using a global identifier (a URI). In order to manipulate these resources, components of the network (clients and servers) communicate via a standardized interface (e.g., HTTP, a protocol that is Client/Server, stateless, cacheable and layered) and exchange representations of these resources (the actual documents conveying the information) [11].

3 IDN Main Views and IDN Naming System

IDN is described through the ensemble of concepts, models and technologies pertaining to the following three views Fig. 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. In the IDN-IM, information is highly structured and endowed of specific metadata. Moreover the Information Model defines the conceptual operations that the applications perform on the pieces of information and documents. Generic information modeled in IDN is formalized as an aggregation of elementary data units, named Primitive Information Unit (PIU). Each PIU contains data and metadata. All data and metadata are handled, by the IDN-Service Architecture. Among those, each metadata element may be visible to applications; it can be defined by the application itself (in this case it is named “application properties”) or by the Service Architecture and, if needed, used by applications. Another class of metadata elements exist for internal Service Architecture use only.

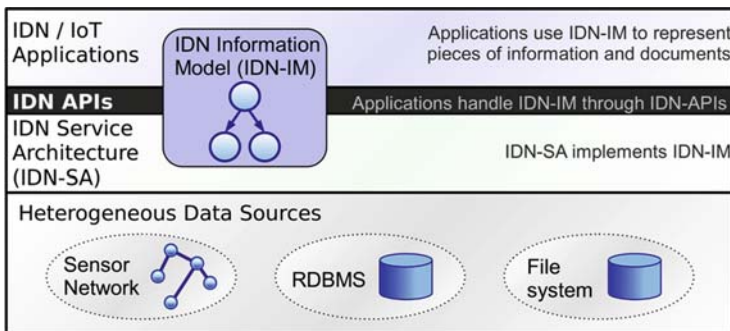


Fig. 1 IDN framework

Each node is addressed by one or more URIs related to one or more names (aliases), as it will be detailed in Section IDN Naming System. An IDN-document structures information units and it is composed by nodes related to each other through directed “links.” Moreover, IDN-documents can be inter-linked, so two 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. Each PIU belonging to the document can also be addressed as a document root node increasing information granularity and reuse. IDN-IM documents express data contents and relation between contents. Data and metadata are structured following the name-value representation and embedded inside the node.

IDN-SA (InterDataNet Service Architecture). It is the architectural layered model handling IDN-IM documents (it manages the IDN-IM concrete instances allowing the application to “act” on pieces of information and documents). The IDN-SA implements the reference functionalities defining subsystems, protocols and interfaces for IDN document management. The IDN-SA exposes an IDN-API (Application Programming Interface) on top of which IDN-compliant Applications can be developed. The IDN-SA encompasses the IDN Naming Systems handling the structured document addressability and resolution of IDN-documents names.

IDN-Applications. IDN-compliant applications implement the context-dependent logic and store/manage information using the IDN-API to exploit the IDN-SA services. IDN applications use the shared IDN-IM to represent and handle their pieces of information and documents.

In accordance to the REST approach, *IDN naming system* adopts an HTTP:URI-based naming conventions to address IDN-nodes. IDN-nodes have, at least, one canonical name which unambiguously identifies the node for each state of its life-cycle; the canonical name has to be globally unique and is assigned once when the node is created. The uniqueness of the name is achieved by the use of URI: the creator of the node has to handle local unique identifiers only, while the LDNS is entitled to do the rest of the job. More than one name can be assigned to IDN-nodes. Such names are also URI-based and are referred to as “aliases.” These are “logical names” of the resource. For instance, the resource describing Temperature Sensor identified by the canonical name:

“<http://interdatanet.example.org/examples/sensors/temperature/tz04b>.”

To refer to the same information in a different context, we can add the alias name: “http://mydomain.example.org/my_home/kitchen_temp.” Starting from these logical names, mechanisms to unambiguously physically locate and access the specific resource are needed. To this end, IDN architecture is composed of three logical conceptual layers (see Fig. 2 left part): (a) *the upper layer handles HFN (Human Friendly Identifiers)* and it is used by applications to identify IDN nodes. The node’s canonical name and its aliases are located in this class; (b) *the middle layer handles URN (Universal Resource Names)* to unambiguously, univocally and persistently identify the IDN nodes independently of their physical locations; (c) *the lower layer handles URL (Uniform Resource Locators)* to identify and access Storage Interfaces adapting the object data model to the IDN-Information Model. As any resource can be replicated in several locations, each URN can correspond to many URLs.

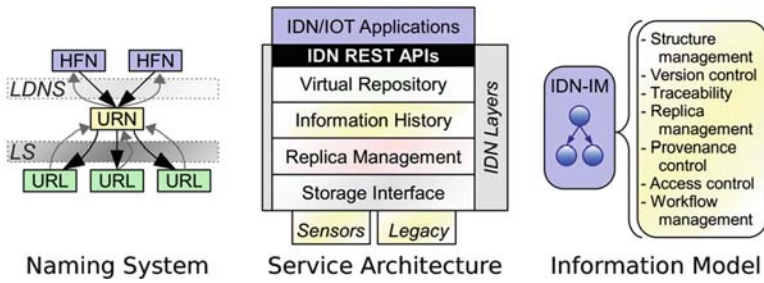


Fig. 2 IDN three layers Naming System, Service Architecture and Information Model

As HFNs, URNs, and URLs are sub-classes of URIs, they are hierarchical and their direct and inverse resolution is possible using DNS (Domain Name System) system [14] and a REST-based approach. By construction, the IDN naming system is compliant with Linked Data principles [4] through the choice to adopt HTTP:URI-based naming conventions and management of identifiers.

3.1 The IDN Service Architecture

The IDN-SA provides an effective and efficient infrastructural solution for IDN-IM implementation. IDN-SA is a layered service-oriented architecture and it is composed of four layers (see Fig. 2, middle): Storage Interface Layer; Replica Management Layer; Information History Layer; Virtual Repository Layer. IDN-SA layers functions are hereafter briefly specified, starting the description from the bottom of the stack. For space constraints, in this section we will not detail on two aspects related to versioning and replica management. Their integration in the IDN architecture is fundamental in order to provide collaboration-enabling functions, but their detailed description goes beyond the scope of the present paper.

Storage Interface Layer (SI). The SI layer provides a REST-like uniform view over distributed heterogeneous data independently from their location and physical data source. The SI provides physical addressability to resources through URLs addresses.

Replica Management Layer (RM). This layer provides a delocalized view of the resources to the upper layer offering URN (Universal Resource Name) which are used here to identify resource, to URL address resolution through the LS (Localization Service). As such the upper layer handles only resources' univocal and persistent identifiers and allows the association of several physical resources (replicas) to the same identifier.

Information History Layer (IH). This layer manages PIUs history allowing navigation and traversing into the versioned information space. At this layer, PIU are identified through URN plus an optional version parameter identifying the time-ordered position. This URI identifies a version of the node.

Virtual Repository Layer (VR). It exposes the IDN APIs to the IDN-compliant Applications exploiting lower layers services. VR provides the maximum abstraction of structured information to the application. Indeed, this layer is seen from the application as the container-repository of all PIUs. The resolution of Human Friendly Names (HFN, logical resource identifiers) into unique identifiers (URN) is realized in this layer exploiting the LDNS (Logical Domain Name System) service which is logically located inside the VR layer (see Fig. 2, middle). A sub-service of the VR layer, namely the Resource Aggregation Service (RAS), is entitled to receive all document requests from the applications. When it receives a request, it collects all the content by traversing the links listed in the PIUs, starting from the root node (addressed through the HFN). Each PIU contributes as a building-block to construct the requested information. In agreement with the request, the RAS applies a conversion in the desirable format and replies.

4 IoT Objects Representations Supported by the IDN Framework

In a typical sensor networks [5], sensor data generated by each sensor node is routed toward a sink node. Then, a system attached to the sink node processes the data format into appropriate data format to be provided to the application [3]. The sink node accepts a query from a client and serves as an interface between the sensor network and applications. The IDN approach does not make any assumptions on the internal of a sensor network other than that the sink node is accessible by the Storage Interface and allows an abstraction of sensor data offering a document representation exposed by IDN-API. The IDN-SI can also be layered into the sink node itself. The IDN-API provides what we name the “IDN Virtual Object” (see Fig. 3) (IDN-VO) document description compliant with the IDN-IM (Information Model) abstracting objects data from implementation details. An IDN Virtual Object can be any kind of data producer, for example, a real sensor, a wireless camera, a desktop computer, a cell phone, or any combination of them.

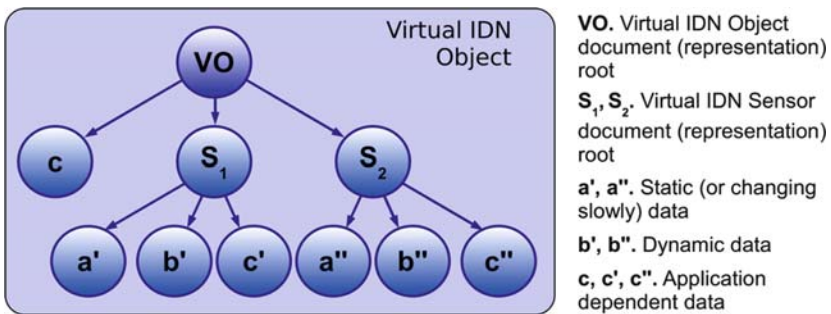


Fig. 3 Example of IDN Virtual Object composed of two sensors (S1 and S2)

An IDN-VO exposed through the IDN-API provides a rich, globally accessible virtual object description composed of three elements: (a) *information related to the object*; i.e., a description of the object itself in terms of parameters which are static (or changing slowly), and it is mainly conforming to the existing standards (e.g., sensorML markup language); (b) *information related to the value/measures performed by the object*; these are couples name-value (e.g., temperatures measures over time), (c) *ad-hoc description of the object created by the application*; this type of information brings into the document model the data which are relevant for the specific application (e.g., metadata for specific smart spaces applications).

The three elements of the IDN-VO are nodes in an IDN document; as such, they are uniquely addressable and can be separately managed by the IDN infrastructure benefiting also from its versioning and replica capabilities. Data handled by the layers of IDN are consequently transformed from low-level data into meaningful, high-level information to be exploited to build applications. Application-level services (such as context-aware services) may use information provided by several sensor networks or interact with actuators located in different networks in order to complete their task. The sensor networks can provide back into the system high-level data or complex actuation tasks that can be used by other application aware services to produce higher level of abstraction or more complex task.

5 Conclusions and Research Challenges

This paper discusses the architectural requirements of a scalable networking architecture that seamlessly integrates heterogeneous information sources to provide rich context aware services. The paper is focused on discussion of architectural elements rather than implementation details. For more complete technical details, see [10] and forthcoming papers.

At the core of the system sits the IDN middleware an infrastructural solution supporting a decentralized and scalable publication space for smart data integration and interoperability. In IDN, the things/object is represented by an IDN document, and IDN Virtual Object, which is a virtual abstraction of the thing (or “virtual thing” [3]) that is interlinked nodes of data and related metadata; it is described in compliancy with the IDN Information Model and it is univocally addressed by an HTTP URI (PRI). Thanks to the properties of the IDN-IM described above, each node of the “virtual thing” is univocally identified and addressed through an URI.

The definition of the IDN Virtual Object, together with the functions provided by the IDN Service Architecture and, in particular, by the IDN Naming System, brings a set of advantages which place the IDN approach beyond the state of the art:

- *The global addressability.* Attained through the IDN three-layers naming system providing scalable and consistent reuse of URI identifiers for reference, addressing, and retrieval of resources on a Web-wide scale

- *The transformation of low-level object information into more meaningful, high-level information, i.e., smarter data.* Attained by the management of the IDN Virtual Objects by the different layers of the IDN Service Architecture
- *The inherent possibility to maximize the efficiency in the creation of “virtual objects”* reusing parts of the objects descriptions of the IDN VO, that is handling the effective orchestration of different objects at an infrastructural level, through the IDN stack, an consequently easing the applications development. Virtual Objects and their data streams can be combined in arbitrary ways, and thus enable the applications to use a data-oriented IoT consisting of an objects networks connected via the IDN
- *The possibility to augment the data management with semantic elements;* this is leveraged by the direct compliancy of the IDN Naming System with the Linked Data [4] approach envisaging the application of the URI naming and HTTP URI addressing to the semantic description of the resources to enable building Web-wide semantic applications [17].

Within the InterdataNet perspective on the Web of Thing (WoT) vision, a number of relevant issues should deserve more deep discussion to validate the approach, however we just mention those that are objects of active research and development in our team. General level issues concern the implementation details of the IDN Service Architecture and Naming System [10], and demonstrating how the IDN fits into the Web of Data and Linked Data view [17]. As for the IDN perspective on the IoT, the following issues are relevant: how IDN controls objects in the IoT, that is how it interacts with actuators, how the IDN makes or enables the discovery of Objects, how the IDN is able to provide notifications from Objects, how the IDN addresses the security issue, how IDN handles real-time availability requirements of some IoT applications, how IDN satisfies the objective to augment its features with semantic capabilities.

Like WWW provided a global space for the seamless integration of local “webs of documents” into a global, open, decentralized and scalable publication space thanks to TCP/IP and internetworking layered solutions, we claim that the realization of the grand vision of the interlinked smart data would be much easier and faster if we could count on an “interdataworking” infrastructure. The IDN layered middle-ware aims to provide an attempt in the direction of an interdataworking vision.

Acknowledgments We would like to acknowledge the valuable support of Prof. Dino Giuli for the material and scientific support to this research activity. Moreover we acknowledge the precious work of Luca Capannesi for the technical support in the development activities.

References

1. AA.VV. (2008) Linked Data on the Web (LDOW2008) Proceedings WWW 2008 Workshop, April 22, 2008 Beijing, China. <http://events.linkeddata.org/ldow2008/>
2. AA.VV. (2009) Linked Data on the Web (LDOW2009) Proceedings WWW 2009 Workshop, April 20th, 2009 Madrid, Spain. <http://events.linkeddata.org/ldow2009/>

3. Aberer K, Hauswirth M, Salehi A (2007) Infrastructure for data processing in large-scale interconnected sensor networks. In: *Mobile Data Management (MDM'07)*, Mannheim, Germany
4. Berners-Lee T (2006) Linked data – design issues. W3C <http://www.w3.org/DesignIssues/LinkedData.html>
5. Culler D, Estrin D, Srivastava M (2004) Guest editors' introduction: overview of sensor networks. *Computer* 37(8):41–49
6. Dolin RA (2006) Deploying the “Internet of Things”. In: *Proceedings of the international symposium on applications on Internet*, IEEE Computer Society, pp 216–219
7. Fielding RT (2000) Architectural styles and the design of network-based software architectures. Unpublished Doctoral dissertation, University of California, Irvine
8. Gronbaek I (2008) Architecture for the Internet of Things (IoT): API and Interconnect. In: *Sensor technologies and applications, 2008. SENSORCOMM '08, Second International Conference on*, pp 802–807
9. Hendler J, Shadbolt N, Hall W, Berners-Lee T, Weitzner D (2008) Web science: an interdisciplinary approach to understanding the web. *Commun ACM* 51(7):60–69. doi: 10.1145/1364782.1364798
10. Innocenti S (2008) InterDataNet: nuove frontiere per l'integrazione e l'elaborazione dei dati. visione e progettazione di un modello infrastrutturale per l'interdataworking. Ph.D. Dissertation, Electronics and Telecommunications Department, University of Florence, Italy, Dec 2008
11. Kawahara Y, Kawanishi N, Ozawa M, Morikawa H, Asami T (2007) Designing a framework for scalable coordination of wireless sensor networks, context information and web services. In: *Distributed computing systems workshops, 2007, ICDCSW '07. 27th international conference on*, p 44
12. Lund K, Eggen A, Hadzic D, Hafsoe T, Johnsen F (2007) Using web services to realize service oriented architecture in military communication networks. *Commun Magazine, IEEE*, 45(10):47–53. doi:10.1109/MCOM.2007.4342822
13. McManus R (2009) Interview with Tim Berners-Lee, Part 2: Search Engines, User Interfaces for Data, Wolfram Alpha, and More... *ReadWriteWeb*, 10 July 2009
14. Mockapetris PV, Dunlap KJ (1995) Development of the domain name system. *SIGCOMM Comput Commun Rev* 25(1):112–122. doi: 10.1145/205447.205459
15. Matthew MacKenzie, Laskey C, McCabe K, Brown F, Peter F, Metz R, Hamilton BA. Reference Model for Service Oriented Architecture 1.0 OASIS Standard, 12 October 2006 <http://docs.oasis-open.org/soa-rm/v1.0/>
16. Pachube: Building a platform for Internet-enabled environments. <http://community.pachube.com/>
17. Pettenati MC, Chini D, Parlanti D, Pirri F. InterDataNet: A Web of Data foundation for the Semantic Web vision *IADIS International Journal on WWW/Internet* 6(2):16–30 ISSN: 1645–7641
18. Prehofer C, van Gorp J, di Flora C (2007) Towards the web as a platform for ubiquitous applications in smart spaces. In: *Second workshop on Requirements and Solutions for Pervasive Software Infrastructures (RSPSI)*, Innsbruck
19. Presser M, Barnaghi P, Eurich M, Villalonga C (2009) The SENSEI project: integrating the physical world with the digital world of the network of the future – [global communications newsletter]. *Commun Mag IEEE* 47(4):1–4
20. Stirbu V (2008) Towards a RESTful plug and play experience in the web of things. In: *Semantic computing, 2008 IEEE international conference on*, pp 512–517
21. Vinoski S (2007) REST eye for the SOA guy. *IEEE Internet Comput* 11(1):82–84
22. Welbourne E, Battle L, Cole G, Gould K, Rector K, Raymer S et al (2009) Building the Internet of things using RFID: the RFID ecosystem experience. *Internet Comput IEEE* 13(3):48–55. doi: 10.1109/MIC.2009.52
23. Zweben SH, Edwards S, Weide B, Hollingsworth J (1995) The effects of layering and encapsulation on software development cost and quality. *IEEE Trans Softw Eng* 21(3):200–208