

# Global Sensor Modeling and Constrained Application Methods enabling Cloud-Based Open Space Smart Services

Anh Lê Tuán, Hoan N. Mau Quoc,  
Martin Serrano, Manfred Hauswirth  
Digital Enterprise Research Institute  
DERI- NUIG, Galway, Ireland  
<anh.le@deri.org>  
<hoan.quoc@deri.org>  
<martin.serrano@deri.org>  
<manfred.hauswirth@deri.org>

John Soldatos,  
Athens Information Technology  
Research and Education Laboratory in  
Information Technologies  
Peania, Greece  
<jsol@ait.gr>

Thanasis Papaioannou and Karl Aberer  
École Polytechnique Fédérale de  
Lausanne - EPFL  
Lausanne, Switzerland  
<karl.aberer@epfl.ch>  
<thanasis.papaioannou@epfl.ch>

**Abstract**— The deployment and provisioning of intelligent systems and utility-based services will greatly benefit from a cloud-based intelligent middleware framework, which could be deployed over multiple infrastructure providers (such as smart cities, hospitals, campus and private enterprises, offices, etc.) in order to deliver on-demand access to smart services. This paper introduces the formulation of an open source integrated intelligent platform as solution for integrating global sensor networks, providing design principles for cloud-based intelligent environments and discuss infrastructure functional modules and their implementation. This paper review briefly technologies enabling the framework, towards emphasizes on-demand establishment of smart cities services based on the automated formulation of ubiquitous intelligence of Internet connected objects. The framework introduced founds on the GSN infrastructure. The framework leverages W3C SSN-XG formal language and the IETF COAP protocol, providing support for enabling intelligent (sensors-objects) services. The service requirements for particular smart city scenario are introduced and initial implementations and results from performed simulations studied and discussed.

**Keywords:** Ubiquitous Intelligent, Smart Systems, Global Sensors Network, Middleware, Intelligent Platforms, Services, Architectures, Smart Environments, Smart City, Data Modeling, Ontologies, Internet of Things, Linked Data, Cloud Computing.

## I. INTRODUCTION

Intelligent systems play a crucial role in the future of the societies and their environment, particularly in the accelerated run for the deployment and provisioning of intelligent smart services. In the era of cloud computing and Internet-connected objects (ICO) or also named Internet-of-Things (IoT) [1]. The deployment and provisioning of intelligent systems and utility-based environments will greatly benefit from a cloud-based intelligent middleware framework, which could be deployed over multiple infrastructure providers (such as smart cities, hospitals, campus and private enterprises, offices, rooms, etc) in order to access ICO data and to deliver on-demand smart services.

In the area of IoT big advances have been realized to enable ICO control (mainly over sensor networks), however full control over multiple devices has not been implemented yet, nor intelligent services deployed.

The design and deployment of intelligent infrastructures enabling smart applications also get big benefits from open middleware/framework (mainly cloud-based), which can be deployed over multiple infrastructure providers [2]. Based on such frameworks, end-users (i.e. individuals, citizens) would be able to leverage towards intelligent providers' sensing and actuating infrastructures, rather than having to deploy sensing infrastructures by themselves.

In the other hand cloud enterprise services, and its emerging close-related technological paradigms [3] should be smoothly integrated within the emerging integrated enterprise services delivery models, for example the Internet-of-Services (IoS). However, despite the proliferation of cloud computing models and infrastructures, there is still no easy ways to formulate and manage Internet connected cloud-based environments i.e. environments comprising IoT "objects" and resources (such as sensors, actuators and smart devices) and dynamically offering on-demand utility-based (i.e. pay-as-you-go) intelligent services. Up-to-date several researchers have described the benefits of a ubiquitous (sensor-based) distributed computing infrastructure [4][5][6] without however providing a systematic and structured solution to the formulation and management of utility-based IoT environments. Similarly, sensing infrastructures and services [7][8][9], provide instantiations of cloud-based and utility-based sensing services (such as the "Location-as-a-Service" in [10]), without however providing any middleware framework and disciplined approach for providing the cited required "intelligence" to the services.

The merits the convergence between cloud computing and IoT have acquired during the last couple of years have given rise the notion of a «cloud-of-things». As result of this popularity many ubiquitous intelligence and smart systems/ environments have been influenced by different definitions and implementations related with cloud concepts, including sensors clouds [11][12], IoT societies and platforms [13][14], as well as social networks of objects [15]. These definitions share common characteristics «things», alike their logical aggregations and virtualization, as well as their participation in business processes. However, this approach does not provide intelligent capabilities in response to end-user requests for smart services per se; furthermore, they lack utility-metering functionalities, and some times other added-value features such as billing, security and privacy.

In this paper we discuss infrastructure functional modules of the framework for enabling the dynamic, self-organizing formulation of optimized IoT applications in cloud environments. This framework aims for enabling intelligence to the services and thus service providers will be able deploy cloud/utility based infrastructures that can deliver IoT services through responding to appropriate end-user requests. The full control of objects is an open challenge, however we claim our approach is funded on standards and extended widely used sensor (physical and virtual) infrastructures which asset principles for a more extended intelligence in the systems towards the full control of ICOs.

The intelligent middleware and its main functional blocks design are researched and partially implemented in the scope of the EU co-funded FP7-287305 OpenIoT project [16], which aspires to create an open source middleware platform that will enable the delivery of non-trivial Internet connected objects applications in a dynamic fashion and according to a cloud-based (pay-as-you-go) utility model.

The paper presents standards and source projects that will boost ubiquitous intelligence and smart environments use cases implementations, along with the main characteristics and building blocks of the framework. The structure of the paper is as follows: Section II presents the main functional components of the OpenIoT framework in the form of design principles and standards-based specifications of the OpenIoT’s approach. Section III, Provides the initial specification of service requests to the OpenIoT platform, by using formal knowledge language and their representations, based on the exploitation of attributes defined in the W3C SSN-XG (Semantic Sensor Networks Ontology). Section IV presents the main characteristics of the GSN [17] infrastructure framework in the form of design principles towards enabling IoT into the cloud. Section V describes the use of constrained applications methodology – linked sensor data modeling to facilitate the linking and combination of the data outputs of sensors and other devices. Section VI presents a use case about ubiquitous intelligence service on smart environments, particularly open space tracking control in smart cities. Section VII presents initial implementation results and initial contributions towards the implementation and experimentation on using GSN and SSN-XG for mobile applications. Finally, section VIII present conclusions of this paper and with references included.

## II. INTELLIGENT OPEN ARCHITECTURE

OpenIoT is a first-of-a-kind design implementation approach of an open-source middleware infrastructure for Internet connected objects (IoT) applications and sensor-based services. OpenIoT serves as a basis for blueprint implementations providing several added-value applications and realizing smart ubiquitous scenarios. General directions in the design process of the architecture are listed as follow:

- Open source developments, through a disciplined licensing and reuse process taking into consideration existing open source projects based on the project partners results and used as based on background technological implementations.

- Experienced and justified design procedures and software tools for formulating communities and mobilizing their members towards extension and contributions to standardization.
- Open source community’s participation and promote track record of successful projects culmination and developing contributions records.
- Produce innovative open source results, which will significantly extend existing results over the RFID/WSN [18] open source implementations.
- Easy adoption of functional blocks as toolkits or libraries to stimulate the interest of existing users and contributors of IoT based developments.

As already outlined, OpenIoT architecture is the provision of an integrated intelligent cloud-based solution for IoT applications and services. Design functionality in the form of functional blocks and operations are illustrated in the figure 1, general descriptions are presented as following.

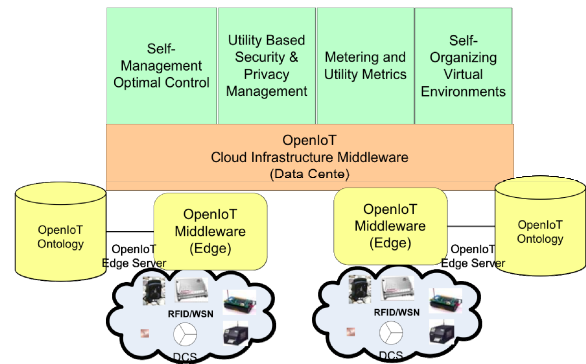


Figure 1. OpenIoT infrastructure functional modules overview.

The OpenIoT architecture is structured in the form of a configurable and integrated “Cloud Infrastructure Middleware” solution for the cloud-based delivery of IoT services. OpenIoT provide the means of formulating clouds of internet-connected objects, which follows deliver utility metering services to end-users models. To this end, OpenIoT research on efficient and scalable ways to process, combine, and enrich data from internet-connected objects. Furthermore, openIoT seek for developing mechanisms for building, deploying, running and delivering applications over the cloud of objects. OpenIoT pursue the idea of enhancing data stream management systems and sensor middleware platforms in order to enable users to deploy sensors/actuators (and other “entities”), as well as to find sensor data provider.

OpenIoT is designed considering the lack of an open source implementation of IoT middleware (beyond early instantiations). Overall, OpenIoT is the result of collective open source implementations of sensor and IoT applications stacks for cloud based delivery. To this end, it will exploit background of other source projects, which have been founded and evolved by several of the consortium partners (EPFL, NUIG, CSIRO, AIT). The availability of these projects, allows OpenIoT to focus on extended research problems at hand (i.e. large scale self-organized cloud based infrastructures) rather than IoT infrastructural problems.

As depicted in figure 2, OpenIoT is by meaning of cloud infrastructure more powerful and the level of orchestration of different GSN instances is considered, it is not difficult to orchestrate instances of GSN working in different cloud environments with the need to exchange data in the form of data streams and result of specific functionalities.

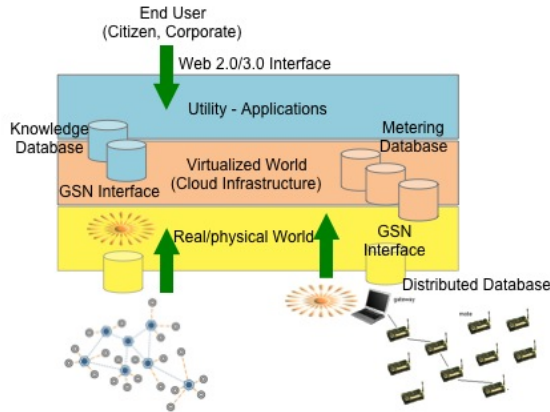


Figure 2. Cloud extensions in GSN.

1. **Self-Management Optimal Distributed Control:** Definition of mechanisms for distributed control of sensing, data processing and actuation. The goal of these mechanisms would be to optimize the value of data gathered and transmitted and to minimize the resources (e.g. energy, bandwidth) spent. The challenge is to realize such an optimization in a large-scale environment.
2. **Utility based IoT services and business models:** Definition of utility-based service provisioning models for IoT applications. To this end, utility models for IoT applications (including ways to measure/meter the utility of the network) are defined in this module, along with tools and techniques for autonomously producing virtualized networks of “entities”.
3. **Security and Privacy Management:** Design and implementation of security and privacy techniques in the various services to be provided by the cloud infrastructure. Particularly methods for expressing security and privacy requirements, along with techniques for decoding these requirements (in the cloud) towards selecting the appropriate security building blocks.
4. **Metering and Utility Metrics:** Support for utility metrics functionalities on the basis of several parameters associated with the fulfillment of an IoT service, such as the number of Internet connected used, the amount of data used, the timeline of the resources utilization, the type of processing algorithms employed and more.
5. **Self-Organizing Virtual Environments Control:** Dynamic self-organization of cloud environments comprising sensors, sensor networks, smart devices and other IoT “entities”. Self-organization of a variety of IoT entities and resources (including ad-hoc), plays a crucial role, mainly because no a physical or a logical address exist as in traditional communication protocols.

6. **Semantic Networks of ICO (Data Centre):** particular focus on semantic interoperability of Internet-Connected Objects, as an enabler of autonomic cloud environments for IoT applications. Extension and if needed development of new ontological representations for IoT “entities”, and enrich them with semantic annotations.

### III. SSN ONTOLOGY & EXTENSIONS

The Semantic Sensor Networks (SSN-XG) ontology (specified by the W3C incubator group<sup>1</sup>) serves as standard base line for specifying service requests on the basis of specific sensors, streams and their properties that should be combined as part of a services request.

The SSN-XG ontology is a good example of a coordinated initiative to specify sensor networks structured data exchange. “The mission of the Semantic Sensor Network Incubator Group is to begin the formal process of producing ontologies that define the capabilities of sensors and sensor networks, and to develop semantic annotations of a key language used by services based sensor networks”.

The Sensor and Sensor Network ontology presented and developed by the incubator group, known as the SSN ontology, answers the need for a domain-independent and end-to-end model for sensing applications by merging sensor-focused (e.g. SensorML), observation-focused (e.g. Observation & Measurement) and system-focused views. It covers the sub-domains, which are sensor-specific such as the sensing principles and capabilities and can be used to define how a sensor performs in a particular context to help characterize the quality of sensed data or to better task sensors in unpredictable environments.

In the other hand the OGC SWE<sup>2</sup> standards provide description and access to data and metadata for sensors, however they do not provide facilities for abstraction, categorization, and reasoning offered by semantic technologies. The semantic annotation method defined by the XG should help the users of OGC standards to retrofit XML-based web services to better support semantic mashups and to ease the integration with linked open data applications relying on semantic web technologies like RDF and SPARQL. The SSN ontology is also the base of the SPITFIRE ontology as described in [19] and is considered in OpenIoT as the base line for semantic sensor networks representation. In parallel activity and as proceed with the framework implementation, each issue that might arise while using the SSN ontology is considered, and it may be proposed to other W3C group members, all the changes we believe are necessary to be applied on the ontology for both revision and extension.

There are necessary extensions of the SSN XG, mainly to orchestrate Internet connected objects (ICOs) in order to have and maintain location-based features, if not already supported on the hardware level, OpenIoT add and maintain location-based properties of Internet-connected objects as

<sup>1</sup> <http://www.w3.org/2005/Incubator/ssn/>

<sup>2</sup> <http://www.opengeospatial.org/>

part of their extension of the SSN XG semantic descriptions. These semantic descriptions are stored as part of the metadata repository located in the underlying cloud infrastructure. Any updates on the location-based properties of ICOs are propagated to their corresponding semantic descriptions in near real-time.

Another necessary extension is to facilitate the inclusion of different types of ICOs on the basis of a variety of wrappers and interconnected objects interfaces. The use of GSN, which does not make any assumptions on the internals of a sensor network other than that a “collector node” called “sink node” is connected to the base computer via a software wrapper conforming to the GSN API. On top of this physical access layer GSN provides so-called virtual sensors that abstract from implementation details of access to sensor data and define the data stream processing to be performed. To support rapid deployment, the virtual sensors in GSN are provided in human readable declarative forms (XML files). OpenIoT aim to provide wrapper at least for the various hardware platforms used (sensors, ICOs) as well as tools and comprehensive documentations to developers for the creation of new wrappers aligned with other technologies.

#### IV. GSN MODELING EXTENSIONS - XGIoT

The Global Sensor Networks (GSN) platform aims at providing a flexible middleware to deal with routing, data aggregation, and energy conservation alike the integration of multiple sensor networks enabling the dynamic integration and management of sensor networks and their produced data streams [17]. Figure 1 depicts the simplest GSN model representation.

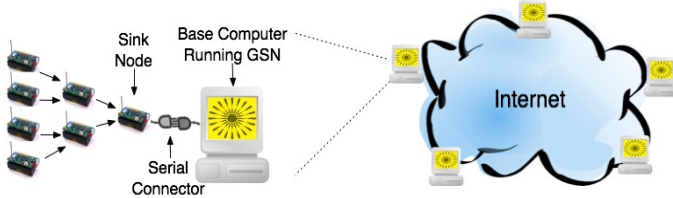


Figure 3. Sensor network data information model in GSN<sup>3</sup>.

GSN assumes the simple model of pipelines through sensors connected to connected sinks sensors: A sensor network internally may use arbitrary multi-hop, ad-hoc routing algorithms to deliver sensor readings to one or more sink node(s) [17]. A sink node is a node which is connected to a more powerful base computer which in turn runs the GSN middleware and may participate in a (large-scale) network of base computers, each running GSN and servicing one or more sensor networks [17]. On top of this physical access layer GSN provides so-called *virtual sensors* that abstract from implementation details of access to sensor data and define the data stream processing to be performed. Local and remote virtual sensors, their data streams and the associated query processing can be combined in arbitrary ways and thus enable the user to build a data-oriented

“Sensor Internet” consisting of sensor networks connected via GSN.

Extensions to GSN are being developed and plans in the original design have been discussed to integrate distributed data streaming repositories where the annotated sensor data will be stored and analyzed alike processed for using cloud infrastructures. The use of a formal language like SSN-XG is necessary when in different cloud domains the data is allocated and shared. Some early design discussions reflect mainly using private cloud and deriving on implementation guidelines. As an example of a sensor query with RDF we have all sensors location within 5km include information about latitude, longitude and the distance to your location

```
Select distinct(?sensorURL) ?lat ?long
<bif:st_distance>( ?geo, <bif:st_point>(-
0.127144,51.506325)) as ?distance
from <http://lsm.deri.ie/metadata#>
where {
  ?sensorURL <http://www.loa-
cnr.it/ontologies/DUL.owl#hasLocation> ?p.
  ?p <http://lsm.deri.ie/ont/lsm.owl#hasCity>
?city;
  <http://lsm.deri.ie/ont/lsm.owl#hasCountry>
?country;
  geo:geometry ?geo;
  geo:lat ?lat;geo:long ?long
filter (<bif:st_intersects>( ?geo, <bif:st_point>(-
0.127144,51.506325),5)).
} order by ?distance
Input: your GPS location
```

Description:

- The <bif:st\_intersects> is the function which is used to find all the sensors around. The bif\_st\_distance is used to calculate the distance to the center point.
- All the sensor metadata (location data) have to be in RDF data and follows the GEO data format.

The graphical representation is depicted in the figure 4, where the located sensors composing the cluster near “My Location” in the example are illustrated, location “n” represent the URL or the associated coordinate to the sensor.

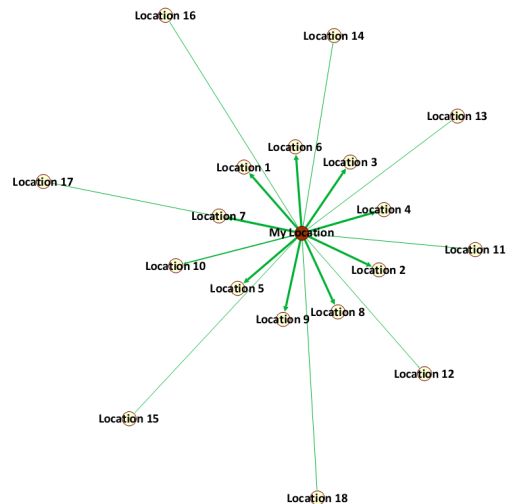


Figure 4. Sensor Cluster Formation based on GSN.

<sup>3</sup> <http://sourceforge.net/apps/trac/gsn/>



## V. LINKING DATA SENSORS AND OTHER OBJECTS

Linked data for information sharing is widely an accepted best practice to exchange information in an interoperable and reusable environment [20], over the Internet different communities use the semantic web standards to enable interoperability and exchange information. Recently linked sensor data has been explored in order to facilitate processes upgrading the processing to a level of sensors gateways, where the data exchange operations can be conducted. To facilitate this process was necessary to identify the persistent problems as listed below:

- In sensor networks only a small numbers of services can be offered, which cannot be personalized to meet dynamic and more composed sensor configurations;
- The offered services in sensor networks are typically technology-driven and static, designed to maximise usage of capabilities of the sensor network as collective services and not to satisfy individual requirements.
- The information from sensors cannot be readily adapted to their changing operational context, sensors cannot communicate their surrounding sensors for configuration purposes, within the objective of service discover and rapid services deployment.

The creation of a data model implies sensor data to be endowed with certain level of information flexibility, operability and management control and most important provide the control platform for interconnected sensors [21].

By using a formal methodology the user's contents represent values used in various service management operations, thus the knowledge-based approach over the inference plane [22] aims to be a solution that uses annotated data to support interoperability and extensibility required in the systems handling end-user contents for more interoperable applications [21]. Beyond of the formal description OpenIoT endows such models with the necessary semantic richness and formalisms to represent different types of information needed to be integrated in network management operations. Early ad-hoc service oriented implementation middleware frameworks enabling the development of complete Service-Oriented-Architectures (SOA) over WSN have emerged such as TinySOA [23], while relevant graphical tools (emphasizing SOA based implementations for WSN) have been also made available [24]. Recently, the rising popularity of RESTful services has also led to the development of RESTful access over sensors and actuators, as is for example the case in the popular PacHube framework [14]. Moving a step beyond service-oriented access, recent works focus on the blending of information stemming from the real-world into business processes. The first true implementations (over realistic enterprise systems) have appeared during the last three years [24], and include the blending of sensor events and processes within conventional Business Process Management (BPM) standards [25][26].

In OpenIoT, a key element is to use Semantic Web technologies to implement applications spanning the Internet and the embedded domain. Hence, our goal is to provide all data through so-called RESTful (REpresentational State Transfer) web services. RESTful web services are services based on HTTP and follow the REST principles as defined by Roy Fielding [27]. The baseline design conception is that in a wireless sensor network (WSN) resources are addressed using standard semantic descriptions in the format of URIs, for example "http://test.com/sensor", where at the same time offering different representation formats e.g., HTML, XML, RDF, thus using HTTP content negotiation and standard HTTP operations such as GET, POST, PUT, or DELETE are used to interact with this resource and to manipulate its state as it is depicted in the figure 5.

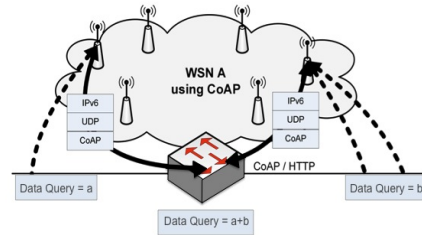


Figure 5. Reach of sensors in WSN by using CoAP.

The use of the Constrained Application Protocol (CoAP), specialized RESTful webs transfer protocol for use with constrained networks and nodes [28], facilitate the exchange of data between sensors and web applications. We have studied two main use cases for using CoAP, the first implies the use of CoAP in combination with DNS (Domain network Servers) for enabling the use of user-friendly fully qualified domain names for addressing sensor nodes. The second do not require the inclusion of DNS but concentrates on M2M communication to avoid unnecessary traffic and thus energy consumption in the sensor network. Both provide a flexible solution to achieve self-organization with a minimum of pre-configuration.

CoAP emerges from the new IETF CoRE working group specifications, which deal with Constrained Restful Environments. CoAP provides exactly the subset of HTTP methods (GET, PUT, POST, and DELETE), which are necessary to offer RESTful web services in a WSN-compatible manner. CoAP provides optional transport reliability, normally a core functionality of TCP, which is due to the resource constraints by nature not available in WSNs. This is particularly useful, since CoAP is designed to be used in combination with UDP which does not offer any reliability but is adequate for WSNs due to its low impact on resources. In addition to the CoAP protocol there is an optional extension (still in draft status) that offers status observations for CoAP resources [29]. UZL and COA have implemented a first prototype version of the current CoAP draft for the iSense hardware platform [30]. In addition, IBBT has implemented a CoAP implementation for the IDRA platform<sup>4</sup>.

<sup>4</sup> <http://idraproject.net/>

## VI. UBIQUITOUS INTELLIGENCE ON RESILIENT SMART CITY ENVIRONMENTS

The information management for cities in cases of unexpected events typically suffers on the availability of up-to-date information. Events like industrial accidents, environmental or medical catastrophes, are requiring the acquisition and distribution of real time information from and to the effected environment. In emerging city infrastructures, there are many potential sources and destinations for this kind of information. Infrastructure objects like bridges can be equipped with sensor devices, people have access to stationary and mobile information devices, cars can be provided with environment sensors, while in emergency situations they could also provide services like battery backed electric power and communication support. In order to enable the on-demand deployment and provision of such smart city services, there is a clear need of a configurable middleware platform that could enable the utility based provision of services.

OpenIoT predicates the formulations of service infrastructures that take into account both live sensor streams and community annotation of environment objects like damaged or blocked roads, emerging of eminent environmental risk. In addition to providing the above functionalities, OpenIoT investigates the possibility of reinforcing resilience through enhancing the redundancy and diversity of sensor information and strengthening the collaborative intelligence and awareness of the communities in smart city environments as depicted in the figure 6.

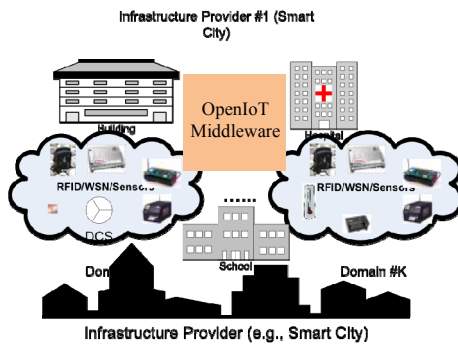


Figure 6. Reinforcing resilience in smart city environments.

## VII. MOBILE SENSING APPLICATION, SIMULATIONS & EXPERIMENTATION

As preliminary research results an example application in the context of smart environments, a service-oriented application named mobile sensing was developed, to test some capabilities defined by the introduced OpenIoT architecture. The mobile sensing application can be seen as part of the self-management optimal control module functionality and the data centre towards enabling the OpenIoT middleware edge support with GSN data being used in across open spaces smart services.

The figure 7 depicts the behavior of three clusters along specific observed data collection time the data collection is by mean of using GSN infrastructure, the IoT extensions and the cluster composition algorithm for linking data sensors.

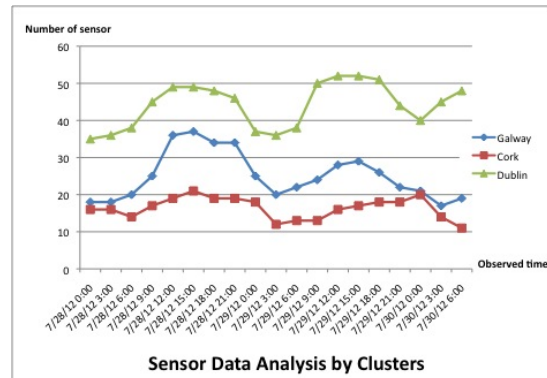


Figure 7. Clustered Sensor Data Analysis.

In this application, people can track crowd behavior by using streaming data and visualize information on a map. As simple as with only pushing the "Map" button, depicted in the figure 8 from the Application home menu GSN sensor data is queried and displayed. Users also can decide if they want to use GSN information as real-time monitoring service, e.g. weather stations, traffic cams, etc.

The various features of the application vary in their data consumption behavior. For example, the live view of the web camera sensor uses larger bandwidth compared to simple map search application. We simulate the various usage scenarios of the application to infer about the amount of data that would be consumed by such applications. Specifically, we model the various types of queries using a discrete random variable and derive the corresponding cumulative distribution using the real time usage of the application. The knowledge of the size of the records for each type of query and the described random variable would let us use inverse transform sampling [31] and obtain the probability distribution corresponding to the data consumption of the application. Thus we would use such knowledge to describe the behavior of the application in various scenarios.

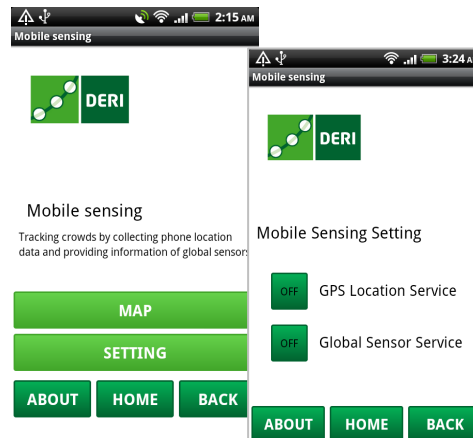


Figure 8. Screenshots from the mobile sensing app main menu.

This application enables users for following groups of people in real-time by graphical representation of location data collected from other users mobile phone as depicted in figure 9.

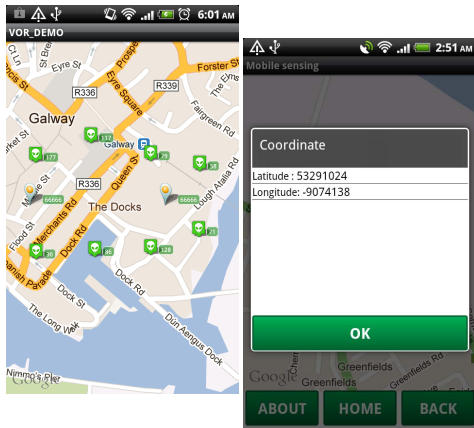


Figure 9. Real-time graphical representation of location data collected from other users mobile phones.

Users are also able to see situation of places around them by providing data from public sensors network (GSN) for example weather conditions on figure 10 or traffic conditions on 11. In this example application a clustering algorithm is used to simplify visualization, depicted in figure 12.

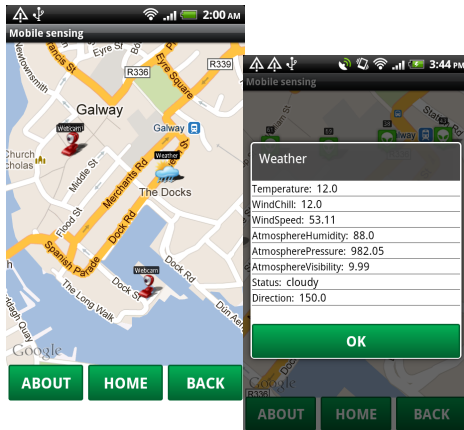


Figure 10. Real-time data collected from weather sensors.

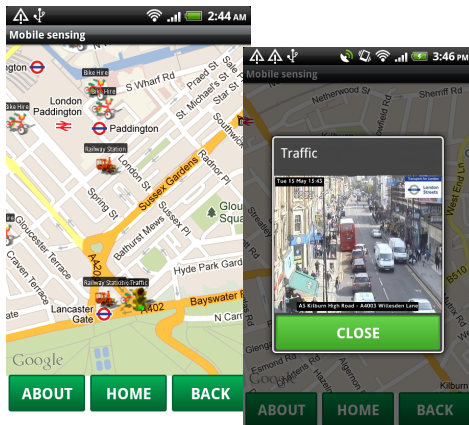


Figure 11. Real-time data collected from traffic cams.

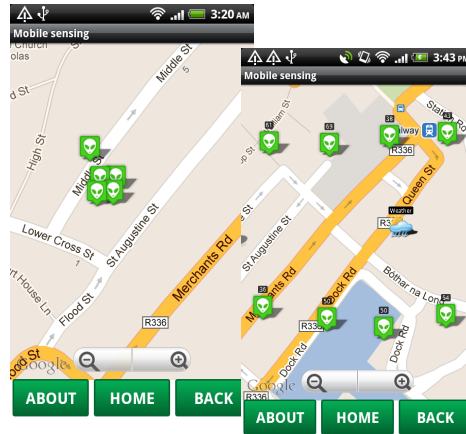


Figure 12. Clustering algorithm used to simplify visualization about big groups of people.

The amount of data consumed by mobile sensing applications on mobile phones is always a major issue. This demands an in-depth analysis of the data consumption behavior of such smart phone applications. We are currently running experiments to analyze data streaming in the SPARQL database, the data from GSN is transformed into RDF streams and by using the mobile sensing application information from the database is collected.

In a restrictive environment such as mobile development where network connections are slow, special attention must be paid to bandwidth usage. In the mobile sensing scenario, the critical task is the fetch of nearby points, especially when the zoom is set to a broad area where the number of results can be outrageous (e.g full city level). We measured the size of each result when applying GZIP compression to lists of more than 1000 entries, obtaining an average of 51 bytes per entry. This means that even by using a slow 3G network of 1Mbit/s, 2,000 results could be transferred in less than a second, which the optimal usage is guaranteed.

## VIII. CONCLUSIONS

In the process of deploying intelligent services the convergence and interoperability of the information plays a crucial role, we have identified requirements and challenges alike introduced the main functional components and operational features of an integrated intelligent platform as solution for integrating mobile and global sensor networks enabling cloud-based intelligent environments.

The intelligent middleware that could support the convergence of utility computing and the internet-of-things has been introduced, based on these design modules. Among the main characteristics of the introduced system is that its implementation could be based on existing open source platforms, which could be enhanced on the basis of popular and emerging standards for IoT.

We have used collected sensor data and by meaning of clustering creation make data analysis for three exemplar data cases. This use case analysis serves as a preliminary implementation, validation and as evaluation environment for the OpenIoT smart city service.

The authors intend to establish a relevant open source project, in order to enable community development of pervasive utility computing based applications and services over cloud environments.

#### ACKNOWLEDGMENTS

Part of this work has been carried out in the scope of the ICT OpenIoT Project (Open source blueprint for large scale self-organizing cloud environments for IoT applications), which is co-funded by the European Commission under seventh framework program, contract number FP7-ICT-2011-7-287305-OpenIoT. The Author(s) acknowledge help and contributions from all partners of the OpenIoT project and to Mario Arias and Vinod Hegde for their help and time to discuss the realization of experiments.

#### REFERENCES

- [1] Harald Sundmaeker, Patrick Guillemin, Peter Friess, Sylvie Woelfflé (eds), "Vision and Challenges for Realising the Internet of Things", March 2010, ISBN 978-92-79-15088-3, doi:10.2759/26127, © European Union, 2010.
- [2] Ching-Hsien Hsu and Hai Jin, "Services Composition and Virtualization Technologies," *IEEE Transactions on Services Computing*, 4(3), pp. 181, 2011. (SCI, EI)
- [3] Michael Armbrust, Armando Fox, Rean Grith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. "Above the Clouds: A Berkeley View of Cloud Computing". Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.
- [4] Hock Beng Lim , Yong Meng Teo , Protik Mukherjee , Vinh The Lam , Weng Fai Wong , Simon See, "Sensor Grid: Integration of Wireless Sensor Networks and the Grid", *Proceedings of the The IEEE Conference on Local Computer Networks 30th Anniversary*, p.91-99, November 15-17, 2005, doi>10.1109/LCN.2005.123.
- [5] Lina Yu, Xiang Sun, Qing Wang, Wanlin Gao, Ganghong Zhang, Zili Liu, Zhen Li, Jin Wang, "Research on Resource Directory Service for Sharing Remote Sensing Data under Grid Environment", *gce*, pp.344-347, 2009 Eighth International Conference on Grid and Cooperative Computing, 2009.
- [6] Kapadia, A.; Myers, S.; XiaoFeng Wang; Fox, G. "Secure cloud computing with brokered trusted sensor networks", *International Symposium on Collaborative Technologies and Systems (CTS)*, 17-21 May 2010, pp.581 – 592.
- [7] Catherine Havasi, James Pustejovsky, Robert Speer, Henry Lieberman, "Digital Intuition: Applying Common Sense Using Dimensionality Reduction", *IEEE Intelligent Systems* 24(4), pp. 24-35, July 2009.
- [8] Nicolas Maisonneuve, Matthias Stevens, Maria E. Niessen, Luc Steels, "NoiseTube: Measuring and mapping noise pollution with mobile phones", *Information Technologies in Environmental Engineering (ITEE 2009)*, *Proceedings of the 4th International ICSC Symposium*, Thessaloniki, Greece, May 28-29, 2009.
- [9] Page, X. and A. Kobsa "Navigating the Social Terrain with Google Latitude". *iConference 2010*, Urbana-Champaign, IL, p.174-178.
- [10] Marshall Kirkpatrick "The Era of Location-as-Platform Has Arrived", *ReadWriteWeb*, January 25, 2010
- [11] Mohammad Mehedi Hassan, Biao Song, Eui-nam "Huh: A framework of sensor-cloud integration opportunities and challenges". *ICUIMC 2009*: 618-626
- [12] Madoka Yuriyama, Takayuki Kushida "Sensor-Cloud Infrastructure - Physical Sensor Management with Virtualized Sensors on Cloud Computing". *NBiS 2010*: 1-8
- [13] Parwekar, P, "From Internet of Things towards cloud of things", 2nd IEEE International Conference on Computer and Communication Technology (ICCT), September 2011.
- [14] Pachube Internet of Things "Bill of Rights" Ed Borden, in *pachube.com* (2011)
- [15] Matthias Kranz, Luis Roalter, and Florian Michahelles, "Things That Twitter: Social Networks and the Internet of Things", What can the Internet of Things do for the Citizen (CIoT) Workshop at The Eighth International Conference on Pervasive Computing (Pervasive 2010), Helsinki, Finland, May 2010.
- [16] EU ICT OpenIoT Project. <http://www.openiot.eu/>
- [17] K. Aberer, M. Hauswirth, and A. Salehi. "Infrastructure for data processing in large-scale interconnected sensor networks". In *MDM'07*, pages 198–205, 2007.
- [18] Nikos Kefalakis, John Soldatos, Nikolaos Konstantinou, Neeli R. Prasad "APDL: A reference XML schema for process-centered definition of RFID solutions". *Journal of Systems and Software* 84(7): 1244-1259 (2011)
- [19] M. Leggieri, A. Passant and M. Hauswirth, "A Contextualized Cognitive Perspective for Linked Sensor Data", in *proceedings at the 3rd International Workshop on Semantic Sensor Networks 2010 (SSN10)*, collocated with the 9th International Semantic Web Conference (ISWC2010)
- [20] Kalinichenko, L., Missikoff, M., Schiappelli, F., Skvortsov, N. 2003, "Ontological Modeling" In *proceedings of the 5th Russian Conference on Digital libraries*, St. Petesburgo, Russia, 2003.
- [21] Hauswirth, Manfred., Pfisterer, Dennis., Decker, Stefan., "Making Internet-Connected Objects readily useful". In *Interconnecting Smart Objects with the Internet Workshop*, Prague, 2011.
- [22] M. Serrano, J. Strassner and M. ÓFoghlú, "A Formal Approach for the Inference Plane Supporting Integrated Management Tasks in the Future Internet" 1st IFIP/IEEE ManFI International Workshop, In conjunction with 11th IFIP/IEEE IM2009, 1-5 June 2009, at Long Island, NY, USA.
- [23] Avilés-López, Edgardo and Garcia-Macias, "TinySOA: a service-oriented architecture for wireless sensor networks", in *Service Oriented Computing and Applications (Springer)*, Volume 3, Number 2, June 2009 , pp. 99-108(10).
- [24] Patrik Spiess, Stamatis Karnouskos, Dominique Guinard, Domnic Savio, Oliver Baecker, Luciana Moreira Sá de Souza, Vlad Trifa: SOA-Based Integration of the Internet of Things in Enterprise Services. *ICWS 2009*: 968-975.
- [25] Serbanati, A., Madaglia C.M., and Ceipidor U.B., "Building Blocks of the Internet of Things: State of the Art and Beyond", 2011. In *RFID/Book 3*, eds. C. Turcu. Rijeka: InTech. (In press.)
- [26] K. Sperner , S. Meyer, C. Magerkurth "Introducing Entity-based Concepts to Business Process Modeling", 3rd International Workshop and Practitioner Day on BPMN, Lucerne, Switzerland
- [27] R. Fielding. "Architectural Styles and the Design of Network-based Software Architectures". PhD thesis, University of California, Irvine, 2000.
- [28] Z. Shelby, B. Frank, and D. Sturek. "Constrained application protocol (CoAP)". <https://datatracker.ietf.org/doc/draft-ietf-core-coap/>, May 2011.
- [29] Z. Shelby, Sensinode, K. Hartke, and C. Borman. "Observing resources in CoAP". <https://datatracker.ietf.org/doc/draft-ietf-core-observe/>, March 2011.
- [30] Coalesenses iSense "A modular hardware and software platform for wireless sensor networks". <http://www.coalesenses.com/isense>.
- [31] Devroye, L. and Devroye, L. "Non-uniform random variate generation", vol. 4, year 1986, Springer-Verlag, NY USA.