EPIKOUROS - Virtualized platforms using heterogeneous sensor services in cloud computing environment

Demosthenes Vouyioukas[2], Athanasios Moralis[1], Manolis Sardis[3], Dimitris Drakoulis[1], George Labropoulos[4], Sofoklis Kyriazakos[4], Dimitris Dres[1]

[1]R&D Department, Telesto Technologies, 62 Imitou st., Holargos, Athens, 15561, Greece
[2]Department of Information and Communication Systems Engineering, University of the Aegean, Karlovassi, Samos, Greece
[3]School of Electrical and Computer Engineer, National Technical University of Athens, Zografou, Athens, Greece
[4]Converge ICT Solutions & Services S.A., 4 Panormou st., Ampelokipi, 11523, Athens, Greece

Abstract—**EPIKOUROS is an innovative project where methodologies and tools will be developed for designing, building and deploying Internet-of-Things applications on Virtual Environments. EPIKOUROS main goal is to design and develop a virtualized middleware platform that would allow easily creating and structuring environments. In turn, it would allow the collection, management and integration of information generated by multiple sensors and sensor-networks, as well as the management of business process procedures that are supported by the sensors' infrastructure. The project will use as proof-of-concept scenarios from independent living, intelligent/multimodal transport and building management.**

Keywords—Internet-of-Things; Virtual Environments; Cloud Computing; Business Process Management

I. INTRODUCTION

In recent years there has been an explosion in the number of applications that utilize multiple heterogeneous sensors and network sensors such as cameras, microphones, temperature sensors, pressure sensors, radio frequency tags (RFID), wireless sensor networks (WSN) etc. [1]. These applications are typically networked with numerous sensors through multiple wireless and wired broadband infrastructures, thus extending internet technologies with sensor technologies and leading to the modern trend of the Internet-of-Things (IoT) [2]. The deployment of multiple sensors and objects can lead to the development of novel applications to a variety of business sectors, ranging from weather forecast, agriculture and transportation to assisted living [3],[4],[5].

Despite the growing number of sensor applications, there are still no mature techniques, methodologies and tools for general purpose design, implementation, integration and finally for the management of multi-sensor applications, such as the development of systems that collect, manage and combine information from multiple heterogeneous sensors. In addition, an sensor based applications usually require complex procedures, business rules in the form of middleware libraries or services, which collect and process information from different sensors, in accordance with the business logic of the developed applications. These applications eventually integrate with other information systems (e.g., Enterprise Resource Planning (ERP), Production Management Systems (PMS), Business Process Management (BPM), Customer Relationship Management (CRM), Manufacturing Execution Systems (MES), etc.) to achieve their business goals.

EPIKOUROS is a General Secretariat for Research and Technology (GSRT) project, co-funded by Greek and EU funds. The project has 30-months duration. Its main goal is to design and develop a virtualized middleware platform that will allow the formation of environments that, in turn, would allow the collection, management and integration of information generated by multiple sensors and sensor networks, as well as the support of business process management infrastructure that are fed by the sensors' infrastructure. Furthermore, the virtualized platform EPIKOUROS will provide an integrated development environment (IDE) that will backup the management of business procedures that consume data from multiple heterogeneous sensors. At the same time, it should allow the integration of these procedures with other information systems (i.e. ERP, CRM, MES etc.).

To this end, EPIKOUROS will provide a development environment that will lead the user to the creation of artefacts for the Software-as-a-Service (SaaS) of EPIKOUROS, translating it in a deployment plan for Cloud Infrastructure. Each service will have a configuration that will be the corresponding artefact and that will be accompanied by a number of instances and connections with other configurations of services. Thus, the hardcoded values assignment and the development of application's versions from scratch are avoided, while achieving interoperability and drastically reducing the development time. Consequently, the development effort is transferred to the business intelligence. Finally, the whole effort will be validated by the creation of four case studies - applications: Fleet Management, Assisted Living, Public Transport Information and Home Automation Application. These applications will utilize the software framework and the SaaS infrastructure that will be developed during the project.

In this contribution, firstly the primary goals of the project EPIKOUROS are presented in Section II and the principal logical architecture is presented in Section III, along with the description of each foremost layer. In Section IV, a detailed description of the prototype services following Service Oriented Architecture (SOA) principles is provided, where the subsystems of the EPIKOUROS platform as services are described in detail and the processes between them are specified. Section V refers to EPIKOUROS roles and the application development procedure, where Section VI entails the cloud deployment of EPIKOUROS focused on cloud management system and the architectural structure of the sensor cloud development. Finally, Section VII concludes the vision of EPIKOUROS project and outlines the work in progress.

II. RELATED WORK AND PRIMARY GOALS

There are a variety of Sensor platforms that allow the integration of sensor hardware and offer the integration service to third parties in order to build their applications. Xively [6] provides a platform for integrating sensors and the tools in the form of client libraries for different systems (C, C++, Java, Rubi etc), including a REST API. A similar approach is offer by Ubidots [7] with the distinction of providing a REST API. Everyware Cloud by Eurotech [8] provides a machine-to-machine (m2m) platform with rule engine, a REST API and the Everyware Software Framework (ESF) for managing data streams. These functionalities are offered over their own cloud premises.

EPIKOUROS will extend on the above platforms by providing:

- User- friendly access to the developer, with the use of simple pervasive Web2.0 interfaces. These interfaces will allow developers to design, configure and manage applications with multiple heterogeneous sensors. They will be internet interfaces (Web2.0/SaaS model) that will not require the installation of complex software at the end-user side. At the same time, the services of the EPIKOUROS platform will be accessible with the use of technical interfaces, independent from the platform's physical location. On a technical and technological level, these interfaces will be based on Simple Object Access Protocol (SOAP) or Representational State Transfer (REST) protocols in a SOA environment.

- Effective use of the charging models based on utility computing models and user's demand. The end-user of the platform's services will be able to determine the parameters of the virtualized environment that will be necessary for the development of applications with multiple heterogeneous sensors. More specifically, the user will provide all the parameters needed for the development of its application, that is to say, the necessary sensors and devices, the hardware/sensor/devices configuration and the necessary software. Respectively, the EPIKOUROS platform will create the appropriate virtualized environment, charging the user accordingly.

- Support of Quality of Service (QoS). Within the framework of the EPIKOUROS platform, the quality of service will be expressed and calculated, not only on the basis of traditional metrics- parameters, (for example, delay, data storage), but also on the basis of metrics related to the sensors environments (e.g., number of sensors or/and wireless sensor networks, supported protocols, supported algorithms for data analysis, charging parameters, etc.).

- Escalation and flexibility, in the framework of several parameters, such as: the number of the supported sensors, supported users, available sensors' databases, sensors' patterns and configuration, etc. The goal of the EPIKOUROS platform is to constitute an ecosystem in which different users will be able to share sensor data (for example, within collaborations or the supply chain management).

EPIKOUROS will offer the required scalability, flexibility QoS and charging mechanism by utilizing Cloud Computing. Specifically, each EPIKOUROS application will be instantiated in the cloud as a Platform-as-a-Service (PaaS) providing a Service-as-a-Service (SaaS) functionality. The platform will provide all the required tools for the management of the Cloud Infrastructure. Figure 1 presents a high-layer view of the EPIKOUROS platform.

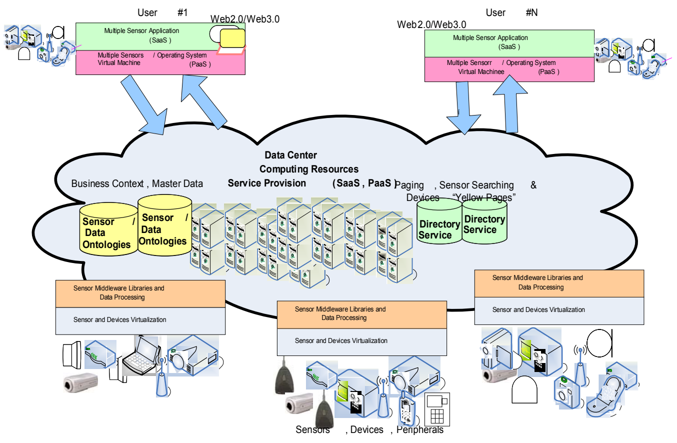
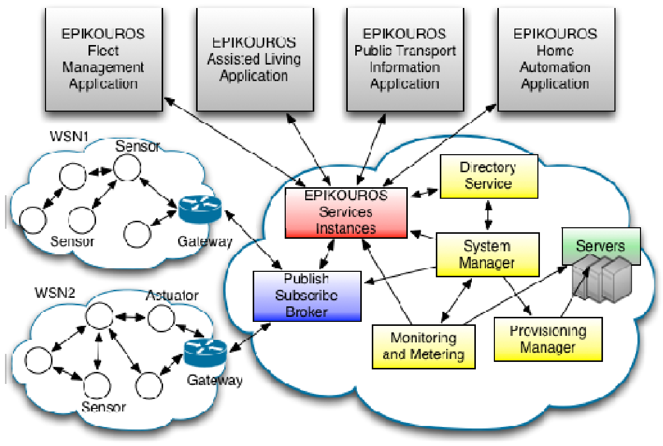

User #1
Web2.0/Web3.0
Multiple Sensor Application (SaaS)
Multiple Sensors Virtual Machine / Operating System (PaaS)
User #N
Web2.0/Web3.0
Multiple Sensor Application (SaaS)
Multiple Sensorr Virtual Machinee / Operating System (PaaS)
Data Center
Computing Resources
Business Context , Master Data
Service Provision (SaaS , PaaS)
Paging Devices , Sensor Searching & "Yellow Pages"
Sensor / Data Ontologies
Sensor / Data Ontologies
Directory Service
Directory Service
Sensor Middleware Libraries and Data Processing
Sensor and Devices Virtualization
Sensor Middleware Libraries and Data Processing
Sensor and Devices Virtualization
Sensor Middleware Libraries and Data Processing
Sensor and Devices Virtualization
Sensors , Devices , Peripherals

Fig. 1. High-level view of EPIKOUROS as part of the Internet-of-Things

III. ARCHITECTURE OF EPIKOUROS

Figure 2 presents the general logical architecture. According to this, the logical layering architecture is defined by Wireless Sensor Networks (WSNs), Cloud Computing and Applications. Sensors and actuators are included in the sensor Physical Layer. The sensors' purpose is to record natural long and short- term phenomena and transform them into a signal that is transmitted to the cloud layer through a wireless sensor network. Respectively, actuators receive a signal from the cloud layer and trigger a natural response (e.g. on/off light switch). Neither the sensors, nor the wireless sensor networks are included in the framework of the EPIKOUROS project, but constitute external systems.

The Cloud Computing Layer contains the subsystems of the EPIKOUROS platform. The application is located in a cloud provider. This application includes a publish-subscribe system (Pub-sub Broker), the cloud's management applications (system manager, provisioning manager, monitoring and metering, mediator, policy repository, collaborator agent, service registry, servers) and finally, the subsystems of the EPIKOUROS platform that are offered as services (Application Specific Services SaaS). These constitute, along with the cloud management subsystems, the basic goal of the EPIKOUROS project.



EPIKOUROS Fleet Management Application
EPIKOUROS Assisted Living Application
EPIKOUROS Public Transport Information Application
EPIKOUROS Home Automation Application
WSN1
Sensor
Sensor
Gateway
WSN2
Actuator
Gateway
Sensor
EPIKOUROS Services Instances
Directory Service
System Manager
Servers
Publish Subscribe Broker
Monitoring and Metering
Provisioning Manager

Fig. 2. General sensor-cloud architectural description in EPIKOUROS

The Application Layer includes the applications that are created by using the EPIKOUROS platform, along with the SaaS services that it offers. These applications use certain SaaS services of the EPIKOUROS platform, create new services that cover users' special needs and provide graphical interfaces, user management, and data models relevant to the solution that the platform provide.

In continuity, the logical architecture of an application based on the EPIKOUROS platform is presented and the general services the platform provides are described. Figure 3 defines the logical architecture of the specific application. An indicative application created on the basis of the EPIKOUROS platform follows a three-tier logical architecture, as the one depicted in Fig. 3. The layers are as follows:

1. **Sensor layer**: This layer includes sensors and actuators that are either autonomous, or placed within a Wireless Sensor Network (WSN). These devices that can be considered as external systems to the EPIKOUROS platform produce data flows in the form of a series of measurements. These data flows can be acquired by the platform using two strategies. The first strategy is based on the publish-subscribe pattern. Following it, the physical sensors send their data encapsulated as messages to the message broker (MQTT), which are forwarded to the driver. This is the usual information flow, particularly for sensors, because sensors, for energy conservation or network configuration reasons, may be in sleep mode and thus not able communicate, or may not have a static address (mobile providers do not provide static addresses). Sensor can be configured with the address that they must send their metrics to. On the other hand, Actuators must be addressable and discoverable, so as to receive mandates from upper layers. They follow the second strategy where the platform through the driver communicates and triggers their physical response.

2. **Service Layer**: It is the layer where the EPIKOUROS platform is offered as a set of services for each application separately. These services are an instance of the prototype services that will be designed and developed within the project's framework. The prototype services are general in nature and their goal is the exposure of sensors as services, the registration and search capability of each sensors' services and the synthesis of the sensor services through service orchestration and business rules.

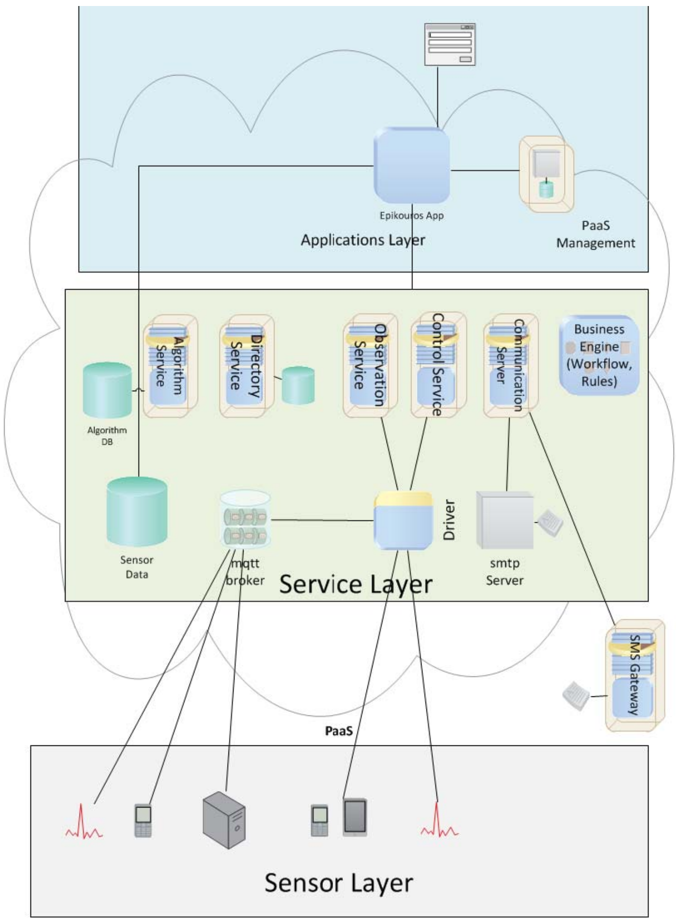


Fig. 3. Logical Architecture of EPIKOUROS in layers

3. **Applications Layer**: This layer includes the business applications that utilize the EPIKOUROS platform for the sensor management in the cloud. These applications have their business uses and implement their own business logic. The logical architecture of EPIKOUROS follows the principles of Service Oriented Architecture (SOA), by defining business operations and capabilities as autonomous, independent atomic services that provide well defined interfaces. These services may be combined through service orchestration in order to create new complex services and applications.

IV. SERVICES PROVIDED BY EPIKOUROS

In this section, we provide a description of the EPIKOUROS's prototype services, as presented in Fig. 3. The subsystems of the EPIKOUROS platform as services following SOA principles are described in detail and the processes between them are specified.

Algorithm service: is the service that manages and provides algorithms for the processing and fusion of multi-sensor data to the rest of the EPIKOUROS based application. These algorithms constitute the implementation of functional filtering units and data combination. As a service, it offers algorithms that perform common sensor data processing functions (e.g. average estimation, variance estimation, etc.) that are useful in various applications. Given an algorithm and defining inputs it returns the computed value. Additionally, it offers the following functions: a) algorithm retrieval, b) new algorithm storage, c) algorithm deletion and d) algorithm search based on specific criteria. Algorithms follow a specific pattern that removes their heterogeneity and allows their integration in the Algorithm Service. This pattern will be extendable in order to assist the implementation of new algorithms for data processing. Algorithms are defined per application and stored in a database.

Directory Service: is the service that provides addressing functions for different sensors in a virtual machine environment. It allows common addressing by unifying different sensor addressing systems (e.g. IP and DNS addresses). It will provide a single addressing mechanism. The purpose of the directory service is to provide addresses independent from the physical medium that have business value. For example, a sensor system, e.g. a device with various detectors, like temperature detectors, light or pressure, located in specific area, could be searched based on these criteria.

MQTT broker: is a commercial-of-the-shelf (COTS) software that implements the publish-subscribe (pub-sub) mechanism for message exchange. The Message Queuing Telemetry Transport (MQTT) is an open Machine-to-Machine (M2M) message exchange protocol between end devices, designed to transport telemetry data in the form of messages. It operates is as follows: each sensor system creates a topic for each evolved detector in the MQTT broker. Then it published the detectors' values on these topics. Upon receiving a new value, the broker starts sending these values to the topic subscribers. The driver subsystem that is responsible for receiving these values registers on these topics.

Driver: is the subsystem whose purpose is to remove the heterogeneity between the sensors and the communication protocols with them. It will support certain communication protocols, like MQTT, but it will be expandable, supporting additional communication protocols. It will convert the received sensor values into an information model common for the EPIKOUROS platform. Additionally, it will be able to send commands to actuators, originating from the Control Service.

Observation Service: is the service that exposes a physical sensor as a virtual sensor. It provides a well-defined interface and is able to uses different service bindings such as SOAP Web Service and Restful Web Service bindings. The type of the supported Web Services depends on the middleware selection for the workflow and rule engine (see Business Engine) for the EPIKOUROS platform.

Control Service: is the service that permits the devices' and sensors' control. This service will allow, through standardized interfaces, the sending of commands from the applications to the actuators.

Communication Server: is the subsystem that permits messages to be sent to third-party entities and external systems. It is provided as a service, following the SOA design of the EPIKOUROS, and will support e-mails through the appropriate SMTP server, and SMS messages through the SMS gateway.

Business Engine (Workflow, Rule): is the main operation of the EPIKOUROS platform, where it implements the application's business logic using workflows, business rules or a combination of both. In the SOA approach that we have adopted for the EPIKOUROS platform, a workflow or a business rule, which combine result from different services, define a composite service.

PaaS management: is the cloud management interface. It permits the management of the EPIKOUROS instance (SaaS) at the runtime.

V. EPIKOUROS Roles and Application Development Procedure

We have defined three basic user roles in the aforementioned architecture, according to the following table, where the roles along with their descriptions are depicted.

TABLE I. Users of the EPIKOUROS project

Role	Description
Sensor Infrastructure Owner	It is the person who manages the sensors and provides their access to the platform based on EPIKOUROS.
Cloud Computing Infrastructure Owner	It is the person who owns and manages the cloud computing and who provides the infrastructure, where the PaaS application is implemented.
Application Developer	It is the person who designs and develops the application that uses the EPIKOUROS platform.

The EPIKOUROS platform is mainly targeted to the Application Developers, offering the tools to design, implement and deploy their application to the Computing Cloud. The Application Developer uses the EPIKOUROS platform in order to develop the application that consumes data flows from the Sensor Infrastructure. Using the provided tools by the EPIKOUROS platform, the Developer can connect the sensors to Observation Services through the Driver component. Then, the Developer implements the application logic using the Business Engine. Finally, using the suitable tools, the Developer can deploy the application to the Cloud of the Cloud Infrastructure Owner. The cloud deployment tools will be general and will support various cloud infrastructures.

VI. Cloud Deployment of EPIKOUROS

As one of the main goals of EPIKOUROS is the seamless integration of the platform tools to the cloud management system, this section is focused on the architectural structure of the sensor cloud development. Accordingly, the EPIKOUROS project will implement appropriate SaaS applications, which can support the exploitation of sensor network's information. For this purpose, a typical PaaS should be enriched with components that can support this development. Although in general the existing solutions, and more particularly the CloudFoundry platform [9], have several components for a typical enterprise application, however they are not suitable for the implementation of these types of applications required in the EPIKOUROS project.

For this purpose, it is necessary to select a PaaS that will allow storage of real-time events, real-time complex event processing, real-time event generation and real-time source recording and event routing. An architecture that covers the aforementioned requirements is shown in Fig. 4. The sensor networks through appropriate adaptors are interconnected with virtual sensor component and then their data into the bus. Also, each adaptor should have a suitable Publish-Subscribe Service interface so as to interface with particular systems, such as sensor networks, in order to develop a virtual sensor.

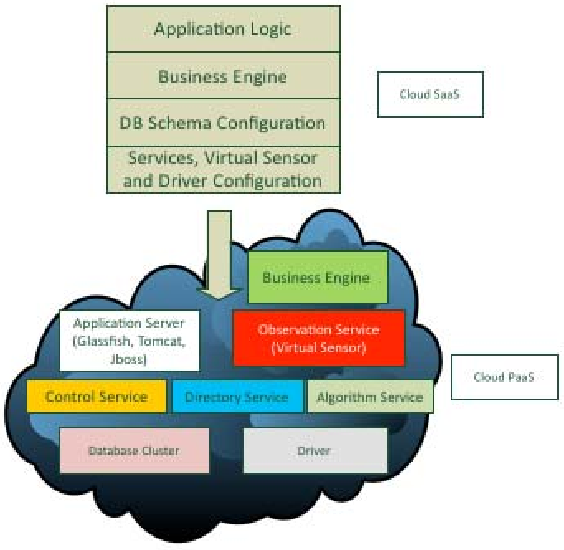


Fig. 4. Cloud Architecture and Deployment of EPIKOUROS

Several components and protocols that the CloudFoundry platform supports will be examined. For instance, the AMQP protocol could support the bus operation and indeed the CloudFoundry platform offers the component RabbitMQ for this purpose. Storing real time events could be implemented as a standard relational database such as MySQL, which is offered by the CloudFoundry or introduced several components like Apache Accumulo, Apache Cassandra, CouchDB, HBase (Hadoop), OpenTSDB, Riak, Redis. Real-time event processing could be implemented through the use of a CEP (Complex Event Processing) system, such as Esper system [10] or Rule/Workflow/CEP Engine, such as Drools [11]. Thus, the CloudFoundry platform could be extended to encompass the above components to support the type of SaaS designated by EPIKOUROS specifications.

The aforementioned components will run on multiple virtual machines where the system's monitoring is essential, so as to make decisions in vertical / horizontal scaling along with re-configuration. Concerning the operation of these logical applications, there are several options that are included in the CloudFoundry platform, such as GlassFish and appropriate BPM systems to run Business Process Execution Language (BPEL) scripts.

Considering the aforementioned attributes, it is clear that an extension of the CloudFoundry platform allows the easily achievement of the main goals of the project EPIKOUROS, minimizing the development of new components, with the exception of the Virtual Sensor as a scalable component with its own challenges. However, the bulk of the work lies in the selection and integration of appropriate components. Thus the corresponding SaaS could be a set of configurations and coding between different components, as shown in Fig. 4.

VII. Conclusions and Future Work

The paper summarizes the main goals of a virtualized platform, called EPIKOUROS, for innovative applications and multiple heterogeneous sensor services in a cloud computing environment. Given that the work is in its infancy, the paper provides the main concepts and ideas underpinning EPIKOUROS, along with a brief description of some of the background work and technologies. The platform is devised to provide methodologies and tools for designing, building and deploying IoT applications on virtual environments and manage the business process procedures supported by the sensors' infrastructure.

During the course of the project, the authors will further develop the PaaS and SaaS components, so as to support the heterogeneity of the multiple sensors, the real-time data processing and the management of the business process procedures. Furthermore, a very promising perspective lies in the use of open cloud platform and applications as instances, providing SaaS functionalities.

Acknowledgment

Part of this work has been carried out in the scope of the EPIKOUROS project, which is a General Secretariat for Research and Technology (GSRT) project, co-funded by Greek and EU funds. The authors acknowledge help and contributions from all the partners of the project.

References

[1] B. Nath, F. Reynolds, R. Want, "RFID Technology and Applications," IEEE Pervasive Computing, Vol. 5, No. 1, pp. 22- 24, Jan.-March 2006.

[2] O. Vermesan and P. Friess, Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems, River Publishers, 2013.

[3] M.-D.Albakour, C. Macdonald, I. Ounis, A. Pnevmatikakis and J. Soldatos, "SMART: An Open Source Framework for Searching the Physical World," Proc. of the ACM SIGIR 2012 Workshop on Open Source Information, 2012.

[4] T. L. van Zyl, I. Simonis, G. McFerren, "The Sensor Web: systems of sensor systems", International Journal of Digital Earth, Vol. 2, Issue 1, 2009.

[5] T. Okadome, T. Hattori, K. Hiramatsu and Y. Yanagisawa, "A real-world event search system in sensor network environments," Proc. Int'l Conf. on Mobile Data Management, 2006.

[6] Xively by LogMeIn, https://xively.com

[7] Ubidots, http://ubidots.com

[8] Everyware Cloud, http://www.eurotech.com

[9] CloudFoundry, http://www.cloudfoundry.com

[10] Esper, http://esper.codehaus.org/

[11] Drools, https://www.jboss.org/drools/