Information as a Service Based Architectural Solution for WSN

Ankur Deshwal, Soma Kohli, Chethan .K.P
SAIT- India
Samsung India Software Operations Pvt. Ltd.
C. V. Raman Nagar, Bangalore - 560 093, India
{a.deshwal, soma.k, c.puttanna}@samsung.com

Abstract— **Wireless Sensors have seen a lot of applications in our daily lives in the recent years. The market has been flooded with high end consumer electronics using wireless sensor technology. However, most of the current technologies require the sensor to be in the vicinity of the end-user application. There has been some study in the techniques for sensor provisioning and sharing for the large number of existing Wireless Sensor Networks. Virtualization of Wireless Sensor Networks (WSNs) is a step forward in exposing these WSNs to large user base from remote locations. However, there is still a huge gap in bringing together information available from heterogeneous, distributed resources of Wireless Sensor Networks to a non-localized user. In this work, we utilize IaaS paradigm of Cloud Computing in virtualization of sensor networks which gives the flexibility of handling heterogeneous systems. The system also enables a smart device user to access information generated by Wireless Sensors through the cloud via SaaS based design. This allows the system to take common computational tasks to be hosted as a service through the cloud. It frees the smart device user from running heavy applications for data processing and storing. Thus, system provides the smart device user a Cloud Enabled Wireless Sensor Network infrastructure. The system architecture provides the necessary features for it to be scalable and flexible, ensuring reliable sensor data transfer and processing through cloud infrastructure. We also present a small test bed implementation of the system.**

***Keywords-* Cloud Computing, Virtual Machines, Sensor Networks, Heterogeneous Systems, SensorML.**

I. INTRODUCTION

In recent years, Wireless Sensor Networks (WSNs) have been gaining increasing attention because of their potential of enabling novel and attractive solutions in areas such as consumer electronics, transport business, health-care systems, environmental monitoring etc. Market study shows that the growth of WSN technology is likely to increase many fold [1]. This may very soon lead to underutilization of available WSNs if the use of current ad-hoc techniques for sensor provisioning and sharing continues. Cloud Computing has, in the last few years, emerged as a new computing paradigm to provide flexible and reliable services which rely on virtualization technologies. Cloud Computing is a model in which computing is delivered to its user as a service rather than a standalone product. It allows user to share and utilize resources. This leads to quick scale-up/down of the acquired resources based on the requirement at that time. Current trends present Cloud Computing in following flavors – Infrastructure as Service (IaaS), Software as a Service (SaaS), and Platform as a Service (PaaS). Since information acquisition systems are becoming important the current research in the rank is "Information as a service" in the area of cloud and wireless sensor technology.

The purpose of our paper is to provide virtualized WSNs which will interact through a cloud infrastructure and provide a mobile smart device user with the required sensor data. To virtualize the sensors, we utilize the IaaS paradigm of Cloud Computing. IaaS provides flexibility in terms of handling heterogeneous resources and ensures high scalability in terms of number of requestors which can share the same resource. Our solution also involves the SaaS in the cloud, which relieves the end user of computational overheads. SaaS also relieves the user from specifying exact sensor characteristics, locating the sensors and provisioning for the sensors. In this paper we propose a flexible and scalable architecture to introduce IaaS and SaaS to virtualized, cloud enabled WSNs. Cloud Computing allows distributing definite responsibilities to different actors in the end-to-end systems. This is imperative in the case of the system discussed in the paper as it involves different actors with different assets and domain knowledge e.g. WSN owners, network infrastructure owners, application writers for data manipulation, end users with smart mobile devices. We discuss briefly in the paper about the roles of each actors of the system which allows it to be a feasible business model.

Rest of the paper is organized as follows: Section 2 describes the related work in the field. Section 3 describes the proposed architecture. Section 4 describes about the roles of actors in the system. Section 5 describes the test-bed implementation of the system. Section 6 concludes the paper followed by reference list.

II. RELATED WORK

Current research in the field of sharing WSNs remotely is focused on virtualizing sensors for provisioning and using them on demand. [2] suggests an infrastructure where WSNs can be configured as virtual sensors. This allows an end user to provision the sensors as and when needed and release them once the requirement is over. The approach is restrictive since it requires the end user to do most of the processing and provisioning. [3] proposes a cloud architecture for integrating WSNs into the cloud. It concentrates on the cloud infrastructure to incorporate the sensors into a global network. However, it does not use virtualization of the sensors. It gives the design of a pub-sub broker in the cloud in detail for channelizing sensor data for SaaS applications. [4] uses the Amazon web services for the same purpose.

GSN ([7][8]) provides infrastructure for abstraction of sensors and is adjacent to our proposed architecture. However, GSN misses the use of current advancements in virtualization for providing abstraction with flexibility, scalability by taking a simplistic approach for sensor provision and accessthrough the cloud.The architecture proposed in our paper is more suitable for real life multi-actor scenarios.

To the best of our knowledge, there is currently no existing “end to end” solution to access data generated from a wireless sensor network located remotely by using a smart device through Cloud Computing. In our paper we provide a solution to this challenge.

III. SYSTEM ARCHITECTURE

The overall system comprises of WSNs henceforth represented as Physical Sensor Networks, a Service Cloud, a Virtual Sensor Cloud and a system for provisioning. The end user requests and receives the services provided by the systems using a mobile smart device. These services are data acquisition, reformatting, processing and delivery of the data from the physical sensors. The overview of the architecture is presented in Figure1.

Different subsystems which comprise the solution are:

a) *Physical Sensor Network.*
b) *Virtual Sensor Cloud.*
c) *Service Cloud.*
d) *Smart Mobile Device.*

The interconnection of above subsystems allows sensor data to flow from sensors system to different end devices requesting it through a scalable and flexible framework. The sensor data is captured by Physical sensor network and passed to Virtual sensor cloud which is responsible for virtualization of sensors. This data is passed to Service cloud where it is processed and finally passed to end devices for display. The subsystems involved are described in rest of the section.

A. The Physical Sensor Network.

Comprises of network of wireless sensors connected with a standard communication protocol (example ZigBee/ Bluetooth).These sensors are connected to a gateway (which can be a small embedded device) which allows the data collected from sensors to be accessible from some computation node outside the network.

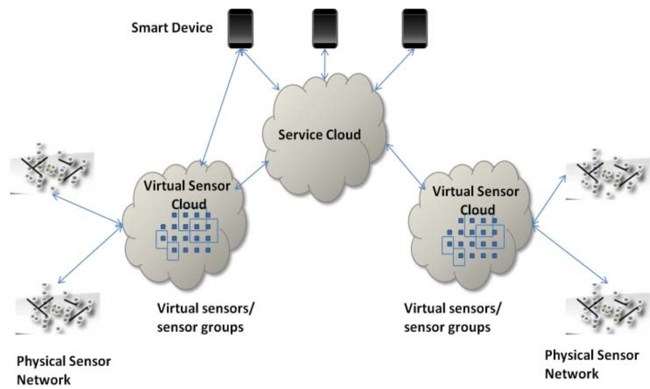

Figure 1 System Overview

B. Virtual Sensor Cloud

This subsystem provides sensors management and sharing mechanism by exploiting IaaS architecture. The sensor data may be requested by many different applications in Service cloud with different requirements. To present the sensor data to the requestor in abstract format, a virtualization layer is required. This subsystem takes the raw data from sensors and provides it to the application in a format requested by the application. Also, to share the sensor with different applications with different requirements simultaneously, a virtualization layer becomes imperative for this architecture. This virtualization layer is composed of two levels for implementation. These levels provide following services-

a) *Abstract representation of physical sensors/sensor networks.*

We will call the abstract representation of physical sensors as “virtual sensors” in the document henceforth. The figure (Figure 2) shows that a group of sensors or a single mote sensor can be exposed as a single virtual sensor. The Physical Sensor Network layer is registered to the Virtual Sensor Cloud subsystem. This provides a one to one connection of the virtual sensor to gateway in the physical layer.

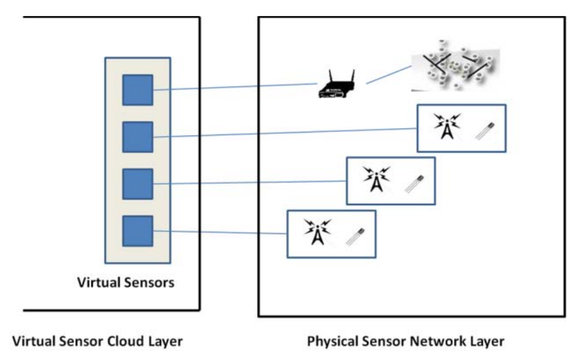

Figure 2 Virtual Sensors and Physical Sensors interconnection

b) Management of Virtual Machines.

Virtual Machines (VMs) are spawned on demand on request from the service cloud. These VMs generate the requested data streams from the raw data received from virtual sensors. These VMs provide data in the format characterized by a template specified using the Sensor Modeling Language (SensorML) specification. The connection formation between a VMs and a virtual sensor follows the Pub-Sub (Publish-Subscribe) mechanism as shown in Figure 3. The responsibility of spawning and managing the Virtual Machines is of the Cloud Controller.

SensorML [5] [6] is primarily used to describe sensor systems and the processing of observations from sensor systems. It is also intended that SensorML provides the framework for encoding the sensor discovery, observations, geo-location information, accuracy information, manufacture information etc. The SensorML speciation provides a functional model of sensors and an XML encoding to describe sensors and their observations. In SensorML specification the XML encoding of SensorML is specified using XML schema. The Figure 3 shows that VM collects measured data from the corresponding physical sensor network port and format the data by considering predefined SensorML.

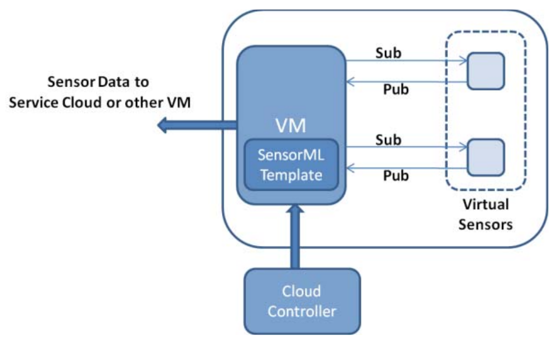

Figure 3 VM Connection to Virtual Sensors

The above discussed layers for this subsystem provide the required virtualization for Sensor Provisioning and abstraction for sensor data. Figure 4 shows an overview of a possible arrangement for the Virtual sensor subsystem. The figure shows a typical scenario where the Service cloud requires data from different WSNs which are not co-located and may be owned by different owners. The Cloud controller spawns one VM at each location which can interact with many different virtual sensors (Figure 4 shows only one virtual sensor per VM). These VMs are categorized as Master VM and Slave VMs. The control and data flow for Service cloud and Virtual Sensor Cloud interaction can be categorized in the following two cases:

a) Connection creation

The Service Cloud requests the initiation of connection (for the receiving data from some sensors) to the Cloud Controller. The Cloud Controller receives the template for the required data format from the Service Cloud, then prepares VM images according to the request and spawns the VMs at the locations where the specified sensors are located. The Cloud Controller configures one VM as the master VM and other VMs as slave VMs.

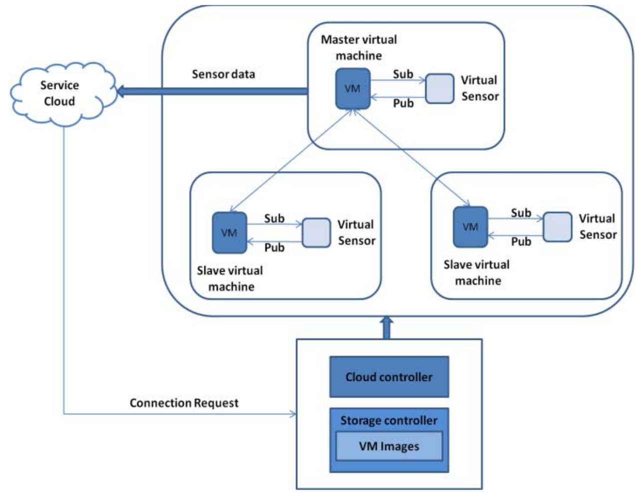

Figure 4 Virtual Sensor cloud overview

b) Data transfer

The VMs subscribe to the data from virtual sensors at each location. The master VM has the additional responsibility of collecting data from other slave VMs, formatting it according to the format requested and sending to Service Cloud. Using the above specified IaaS based architecture with virtualization has many advantages over using ad-hoc process and thread based solution for sending data to the Service Cloud. The comparison of these two solution approaches is as follows-

a) Platform independence- The gateway specified in Physical Sensor Network subsystem can be of different capabilities and may be running different platforms. The platform on which the data collection/forwarding is being done should be abstracted from the user who wants to create the logic for collecting data from virtual nodes. This allows users to use different sensor networks flexibly. Virtual machines are the way to abstract the platform on which the actual logic for data collection, manipulation, and forwarding happens. This allows the developer to develop a single logic and deploy it over all heterogeneous resources available. Using processes and threads for data collection and forwarding leads to requirement of recompilation/porting of the logic for different platforms.

b) Load balancing –

In a cloud enabled WSNs scenario, the load is likely to be uneven and highly dynamic. There is a high chance of a network not being of much interest for a long time and becoming heavily loaded with data requests at other times. VMs based approach allows the load to be balanced by allowing dynamic movement of VMs over different systems [9] [10]. Also, a change in VM role (i.e. Master/Slave) for data movement and manipulation is possible and is much easier. Processes and threads require special assistance for load balancing.

c) Efficiency

High computability and power efficiency is a very favored requirement for such systems. Processes and threads are generally more compute efficient than running a VM. Currently, VMs used in Cloud Computing are generally System VMs due to their flexibility. These VMs are heavier than processes and threads due to dynamic binary conversion required for whole system [11]. However, the architecture allows the use of Process VMs (like Java VMs and Dalvik VMs) to be deployed instead. This brings the efficiency of VMs near to process/thread based solutions without sacrificing the benefits of VMs discussed above.

C. Service Cloud:

This subsystem provides the applications available as SaaS for data manipulation. The major utility of this subsystem is to allow flexible reuse of common operations on data provided by the sensors and transfer of computation intensive applications from power critical end devices to less power sensitive computing environment. These applications do the required operations on data acquired from the Virtual Sensor Cloud and send it to end applications for display. The SaaS in service cloud can be used for two types of services as detailed below –

a) Generic data manipulation services

These include services that provide the common operations on the sensor data e.g. data aggregation services, data reformatting services, and data storage services. These services are required in most of the data processing applications. Flexibility and resource utilization can be increased by making them available as a common service will increase.

b) Application specific services-

These services include the applications developed for data manipulation obtained from sensor networks. An example is a weather forecasting service which takes environmental data as input to provide weather forecasting.

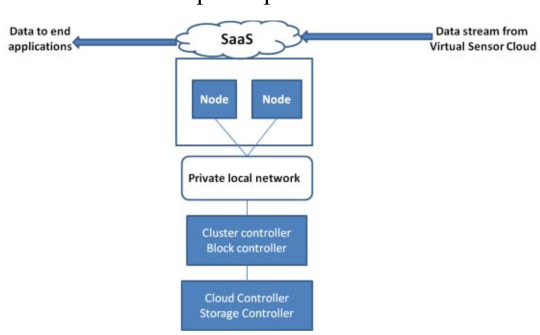

Figure 5 Hosting SaaS application over private cloud.

Hosting an application in SaaS can be done on public clouds through vendors providing SaaS services. Although, in case SaaS application developer wishes to host the service on private cloud, it can be done by creating an private virtualization based infrastructure and hosting the application in a way which allows the load on the application to be dynamically shared among the resources of IaaS. Figure 5 shows the arrangement of private cloud (architecture of IaaS derived from UEC based solution)

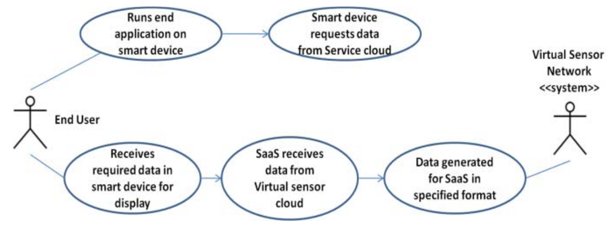

Figure 6 Data acquisition by end user

IV. ROLES AND RESPONSIBILITIES

This section discusses the roles and responsibilities of different actors which are involved in the deployment of the system. The primary actors are – WSN owners, IaaS vendors (for hosting SaaS over private clouds), application writers for hosting applications as SaaS, end users with smart mobile devices. These actors interact with each other and system actors to create a flexible system which can be modeled for monetizing the services transparently with predefined responsibilities.

Figure 6 shows the use-case diagram which captures the relationship of the End User and Virtual Sensor Cloud provider. The End User starts using the application with his smart device which connects to the SaaS layer. After the SaaS finds out which virtual sensors are needed for serving the requests from the application, it requests the Virtual Sensor Cloud to create the infrastructure for the same. After all required connections are established, the user starts receiving intended data from SaaS to his smart device. This diagram captures the ease of operation for the end user and SaaS developer due to virtualization of the sensor network. the use case diagram which captures the relationship between the SaaS developer and IaaS administrator for development and deployment of SaaS application in the Service Cloud system. SaaS developer uses the development tools to develop the software and host it as a SaaS. These tools can be stand- alone tools which can build the applications with SaaS design or these tools can itself be available as PaaS (Platform as a Service). Such tools hosted as PaaS will have the advantage of easy integration with the SaaS layer.

The IaaS administrator has the responsibility of configuring IaaS over which SaaS runs. SaaS is required to be hosted over IaaS to allow fast scale up/down according to requirement. For public clouds, the IaaS administrator will be a service provider which allows SaaS hosting over its own infrastructure. The SaaS developer has to contact the IaaS administrator for hosting the SaaS.

Figure 7 shows responsibilities of a WSN owner for registration of its WSN to be used by end users via the Service Cloud. It also shows the responsibilities of the IaaS provider for the Virtual Sensor Cloud. This IaaS provider can be different from the Service Cloud IaaS provider. An IaaS administrator here has the responsibility of taking the specifications of the WSNs available from the WSN owner in SensorML format and set up infrastructure to provide virtualization for the WSNs. This involves connecting the machines (with required processing power to allow VMs to run) to the gateway made available from the WSN owner. He/She is also responsible for creating customized VM images and launching new VMs when new connection requests come in with their own specifications. Also, load balancing for maintaining high QoS is also the responsibility of this IaaS administrator. Once VMs have been launched, it provides data directly to the Service Cloud.

V. IMPLEMENTATION

We have developed a proof of concept working demo for each of the services described in the solution. In this section we briefly describe the details of our setup and functioning of our test-bed implementation for the system described.

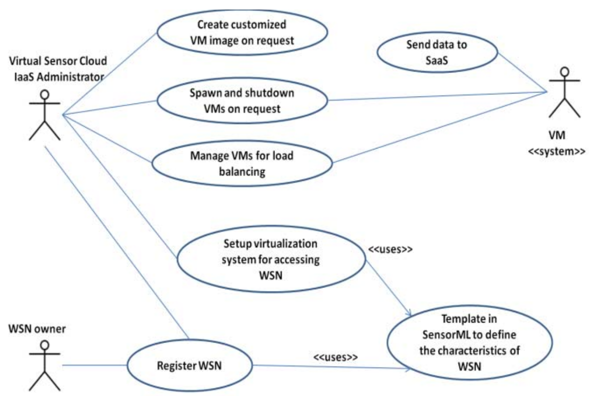

Figure 7 WSN sharing via virtualization

A. *Wireless sensor network:*

We used two kinds of sensor networks to show the wireless sensor network and their integration with the virtual sensor cloud are detailed below –

1) *WSN using TOSSIM simulation:*

We used Tiny OS simulator [12] for the simulation of WSN which has 100 sensor nodes, with the help of TYMO protocol [13] we implemented a mesh topology which has maximum hop count of five and each node in the network are programmed to send a simulated sensor data whenever there is an event, here the event is randomly generated by the each sensor node. In this simulation the sink node has an ability of sending gathered data to other system using TCP/IP or other communication protocol. This sink node sends event data received from the nodes to VSC1 via TCP/IP (VSC1 from the Figure 8).

2) *WSN using TI sensor node:*

Another implementation used CC2530ZNP kits [14] from (Texas Instruments). Each set is a Zigbee network processor mini development kit which consists of three target boards which allowed us to set up a small sensor network. Two battery boards act as mobile sensor node; the sensor boards include an accelerometer, a temperature sensor, and a light sensor. The third board is connected to the Virtual Sensor Cloud PC (VSC1 from the Figure 8) through USB, which acts as a receiver and network coordinator. These boards can be programmed using IAR kick start IDE and supports 'C' as coding language.

B. *Virtual Sensor Cloud:*

We used two Linux systems to implement this subsystem. One system acts as the Cloud Controller (VSC2) and other system acts a node on which VMs can be run (VSC1).

Figure 8 shows the arrangement of Virtual Sensor Cloud and its connections to the physical sensor network subsystem and the system which has the simulated WSN. The Coordinator node for wireless sensor networks setup receives the sensor data from remote nodes and passes it to the VSC1. VSC1 acts as the gateway as well as the node to run VMs. The Virtual Machine Monitor (VMM running on this system) converts the USB interface to the virtual COM port. This virtual COM port acts as the first level of virtualization to the physical WSNs. In this implementation, each virtual COM port available acts a virtual sensor which represents the WSN itself. If finer granularity is required, e.g. one virtual sensor is required corresponding to each physical sensor, hypervisor can create many virtual COM ports with each forwarding data of single sensor.

Sensor data exposed by these virtual COM ports is read by VMs from the virtual sensors. VMs can process the data and transform in requested format. VSC2 act as the (Open Nebula) cloud controller.

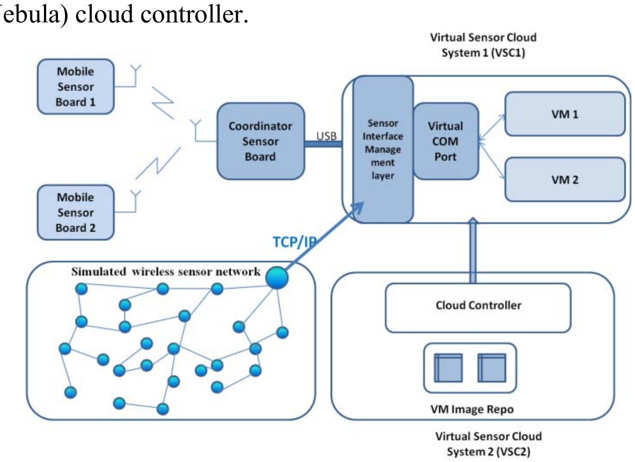

Figure 8 Physical Network and Virtual Sensor Cloud implementation

C. *Service Cloud:*

This part of the system is concerned about hosting the application in SaaS format. A SaaS developer has a choice to use either PaaS tool or stand- alone tools to create the SaaS. Also the IaaS for running SaaS can be chosen to be a private cloud or public cloud. We chose the platform Google App Engine for creating SaaS. This allows flexibility in many fronts. It allows hosting the server for deploying app (application hosted as Google App) locally and on Google public cloud. This allows user to use private cloud or public cloud for deployment. Also, Google App Engine allows the flexibility to use Python, Java and go programming languages for developing SaaS.

D. Smart Mobile device:

We have used the browser over the Android simulator as the end application. We were able to access the average values and graph from the service provided by Google App Engine based Service Cloud.

E. Data flow in Clouds:

Data flow through the entire system which takes place on a request from an End User is shown in the form of a sequence diagram in Figure 9.

Let us consider the example of a temperature sensor based application for explaining the diagram. The App takes the temperature as input at a specific sampling rate from the Sensor Cloud and provides an average temperature (over a period of time) as the output to the end applications. Since the Service Cloud app has subscribed to the temperature data from the temperature sensor in the Sensor Cloud, the Sensor Cloud sends temperature data in SensorML format to the Service Cloud in a HTTP POST request form. The request handler provided with the App engine accepts the request and passes the temperature info to computation and storage engine which calculates the average including the new data and stores the data. Now consider an asynchronous request for current value of average temperature from an end device to the app in HTTP GET format. The result (average temperature) is read from the computation engine and response is created in HTML format. This result is sent to the end device which displays the data using browser.

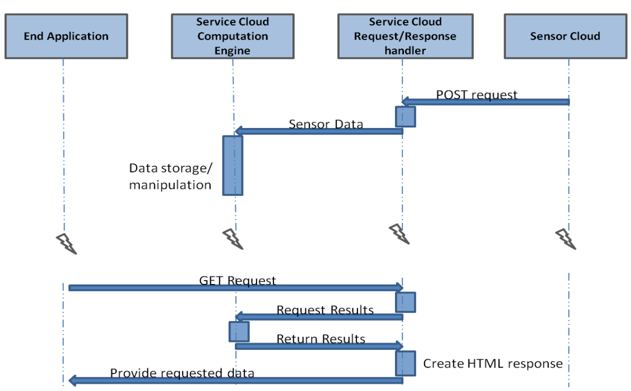

Figure 9 Sequence diagram for data flow on request from end user

VI. CONCLUSION

In this paper we have presented an end-to-end solution for accessing the Wireless Sensor Network on a Smart Mobile device through the cloud. Using this architecture each sensor/sensor group of WSN can be accessed by the user via the cloud. This cloud based architectural solution provides a flexible way of handling a large numbers of heterogeneous sensor devices and sensor. This architecture also provides data virtualization for the WSN. A test bed implementation helped us to provide proof of concept for proposed architecture. There is further expansion scope in terms of load balancing, providing universal abstraction for all class of sensor/sensors network and improving QoS of the overall system.

REFERENCES

[1] In Double digit growth of MEMS and Sensors market report, http://www.isuppli.com/MEMS-and-Sensors/ MarketWatch/Pages/MEMS-Industry-Needs-New-Killer-Apps-to-Sustain-Double-Digit-Growth.aspx, (May 2, 2012).

[2] Yuriyama, M.; Kushida, T.; "Sensor-Cloud Infrastructure - Physical Sensor Management with Virtualized Sensors on Cloud Computing," *Network-Based Information Systems (NBiS), 2010 13th International Conference on* , vol., no., pp.1-8, 14-16 Sept. 201

[3] Mohammad Mehedi Hassan, Biao Song, Eui-Nam Huh; "A framework of sensor-cloud integration opportunities and challenges",, *Proceedings of the 3rd International Conference on Ubiquitous Information management and communication.,* ICUIMC '09

[4] Lee, k.; Murray,D; Hughes,D; Joosen,W. ; "Extending sensor networks into the Cloud using Amazon Web Services," *Networked Embedded Sustems for Enterprise Applications*(NESEA),2010 IEEE International Conference on, 25- 26 Nov, 2010

[5] SensorML introduction http://www.ogcnetwork.net/SensorML (May 2, 2011)

[6] Giovanni Aloisio , Dario Conte , Cosimo Elefante , Gian Paolo Marra "Globus Monitoring and Discovery Service and SensorML for Grid Sensor Networks" 15th International on workshop WETICE'06.

[7] Karl Aberer , Manfred Hauswirth , Ali Salehi, "Infrastructure for data processing in large-scale interconnected sensor networks " Mobile Data Management (MDM), Germany, 2007.

[8] Ali Salehi, Mehdi Riahi, Sebastian Michel, Karl Aberer." Middleware for Stream World " 10th International Conference on Mobile Data Management (MDM 2009), May 18-20, 2009, Taipei, Taiwan

[9] Satyanarayanan M, Bahl P, CaceresR,Davies N,"The Case for VM-Based Cloudlets in Mobile Computing" Pervasive Computing, IEEE Volume: 8 Issue:4 page(s): 14 - 23 Date: Oct.-Dec. 2009 .

[10] sapuntzakis, c.,chandra, r., pfaff, b., chow, j., lam, m., rosenblum, m. Optimizing the Migration of Virtual Computers. In Proc. of the 5th Symposium on Operating Systems Design and Impl. (Boston, MA, Dec. 2002).

[11] Jim Smith and Ravi Nair. 2005. Virtual Machines: Versatile Platforms for Systems and Processes (The Morgan Kaufmann Series in Computer Architecture and Design). Pages 13 – 17. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[12] Özarslan, S., & Erten M, "Simulation of Agilla middleware on TOSSIM" ICST, vol 66, no 6, pp.66.1-66.6, 2008 .

[13] Romain Thouvenin "Implementing and Evaluating the Dynamic Manet On-demand Protocol in Wireless Sensor Networks " June 11, 2007

[14] CC2530 ZigBee Network Processor Mini Development Kit http://focus.ti.com/docs/toolsw/folders/print/cc2530zdk-znp-mini.html#descriptionArea (August 16, 2011).