An Efficient and Extensible Service based Sensor Access Architecture
R.S.Ponmagal
Department of CSE, Dr.MGR Educational and Research Institute University, Chennai, India
rsponmagal@gmail.com

Abstract - This paper aims to design and implement an efficient service based sensor access architecture for web based interaction of sensor nodes. The future sensor networks are envisioned as comprising heterogeneous devices assisting to a large range of applications. Interoperability is required for such heterogeneous devices. To achieve this, Service Oriented approach for the data acquisition from sensor network, and an extensible architecture in which this web services based deployment is extended to CLOUD. The data acquisition from sensor network is done in which sensor nodes are service providers and applications are clients of such services. The service oriented architecture for sensor network has been implemented in such a way that for every specific requirement of the client, the services of the sensors are invoked through a registry and the specific changes in the sensed parameters are also notified as auditable event using push interaction pattern of SOA. Hosting sensor node’s data as web service challenges battery life, bandwidth, processing power constraints of low power sensor nodes. The designed based Gateway reduces the power consumption of sensor nodes hence maximizes the network lifetime. The main goal is to design a flexible and efficient architecture in which sensor networks data can be accessed by applications.

Keywords – architecture, cloud, gateway, sensor, SOA, web service.

1. INTRODUCTION

Wireless Sensor Networks consists of energy constrained sensor nodes and a Sink node with higher processing capabilities. The sensors are physically composed of electronic sensing circuitry, a processor and a wireless transceiver, plus a power supply unit (battery). Sensor networks are distributed event based systems that focus on simple data gathering applications and operate notably differently from that of traditional computer networks. The gathered data can be made accessible to other nodes, including a specialized one called sink through a variety of means. The future sensor networks are envisioned as comprising heterogeneous devices assisting to a large range of applications. Interoperability is required for such heterogeneous devices. To achieve this, In this paper a model is presented, which combines the concept of wireless sensor networks with the Internet through the Gateway.

Here the sensor nodes are service providers and applications are clients of such services. Hosting a web service challenges battery life, bandwidth, processing power constraints of low power sensor nodes. The language in which the sensors are proposed to speak is Quantized xml for web services. The Quantized xml specifies a lightweight subset of the overall Web services protocol suite that is appropriate for network-connected sensors. The Quantized xml for web services reduce the data processing at sensor nodes, while keeping the complex data processing at sink. The Quantized xml
reduces the power consumption of sensor nodes, hence maximizes the network lifetime. Complete set of functionalities for sensor integration and limited constrained functionalities of sensors can be specified. It further reduces the interdependencies between the sensors. Complete set of functionalities for sensor integration and limited constrained functionalities of sensors can be specified. It further reduces the interdependencies between the sensors. The huge amount of data, which a sensor network is able to deliver, demands a powerful and scalable storage and processing infrastructure. Depending on the sample frequency (e.g. from 100 Hz or more down to few samples a day for calculating observations) of the sensors, the deployed infrastructure has to scale up memory, storage and processing power.

Today wireless sensor network platforms e.g. TOSSIM, Crossbow MicaZ, Sentilla JCreate, SunSpot, that perform sensing and complex calculations are most of the time constrained in their capabilities and therefore is one appropriate way to solve this issue is to do offline processing.

Further an architecture is presented, which combines the concept of wireless sensor networks with the cloud computing paradigm, and show how both can benefit from this combination. Sensor data access is thus moved from loosely managed system to a well managed Cloud. The integration of sensor information into the cloud through the sensor as a service through sensor profiles for web services languages will prove the faster communication establishment. The scalability of this approach seems to be unlimited, since wireless sensor networks operate independently, and are connected to the cloud computing environment through a scalable number of wireless sensor network communication gateways. The cloud computing environment itself offers a scalable infrastructure, which makes it very attractive. Hence, the sensed information is deployed into the CLOD-BEEZ, cloud architecture. The combination of wireless sensor networks, with their huge amount of gathered sensor data and their limited processing power, with a cloud computing infrastructure makes it attractive in terms of i)integration of sensor network platforms from different vendors, ii) scalability of data storage, iii) scalability of processing power for different kinds of analysis, iv) worldwide access to the processing and storage infrastructure) resource optimization, vi) be able to share the results more easily, and vii) using pricing as one more criteria for the IT infrastructure.

The present work defines the proposed architectural components as well as the Quantized xml middleware. The paper is organized as follows: Section II covers the state of the art. Section III describes the related work. Section IV,V,VI and VII details the system architecture and the related components. Section VIII shows the performance analysis. Section IX outlines the conclusion.

2. STATE OF THE ART

The sensor information can be transmitted to the requesting client as [1] SOAP messages, which is used to access the sensed information with application independent protocol. A service approach for the design of wireless sensor networks is explained. Services are defined as the data provided by sensor nodes and the applications to be executed on those data. Clients access the sensor network by submitting queries to those services. The DPWS proposal is optimized as TinyDPWS [2] with application specific format technique, reduces the energy consumed by the sensor nodes. This is achieved by using the proposed middleware which provides to
the wireless sensors a Service Oriented Architecture connection to the Internet. The proposed middleware is based on the Device Profile for Web Services which is a Service Oriented Architecture technology at the device level.

A method to access the sensor information using structured data [3] and WSDL descriptions is proposed. The functionality and data provided by the new nodes is exposed in a structured manner, so that multiple applications may access them. The result is a highly inter-operable system where multiple applications can share a common substrate. A key challenge in using web services on resource constrained sensor nodes is the energy and bandwidth overhead of the structured data formats used in web services. Integrating wireless sensor networks in heterogeneous net-works is a complex task. A reason is the absence of standardized data exchange format that is supported in all participating sub networks. XML has evolved to the de-fact standard data exchange format between heterogeneous net-works and systems.

However, XML usage within sensor networks has not been introduced because of the limited hardware resources. In this paper, an XML tem-plate objects are introduced making XML usage applicable within sensor networks. Different optimized way [4] of using XML is specified. This new XML data binding technique provides significant high compression results while still allowing dynamic XML processing and XML navigation. The standard device profiles for web services [5] which could be used for wireless sensor networks is proposed. Even if DPWS is the best candidate to integrate WSN in existing infrastructures, it cannot be applied to WSN without research efforts, because it addresses softer resource constraints as required in WSN. But DPWS provides a minimal set of constraints for applications in resource constrained devices. This paper describes an approach that further restricts DPWS for WSNs, but keeps it still interoperable with DPWS. Several service discovery protocols for wireless sensor networks [9] are proposed.

![Sensor Network architecture](image)

**Figure.1 Sensor Network architecture**

In addition, to reduce power consumption we presented an activation schedule, based on the mapping of the nodes' operational modes to Bluetooth states. By announcing the activation schedule as a service, are presentation of the state of the nodes is exposed to client applications.

A cloud storage platform [6] for pervasive computing environments such as wireless sensor networks is explained. Data storage and sharing is difficult for these sensors due to the data inflation and the natural limitations, such as the limited storage space and the limited computing capability. Since the emerging cloud storage
An Efficient and Extensible Service based Sensor Access Architecture

solutions can provide reliable and unlimited storage, they satisfy to the requirement of pervasive computing very well. Thus a new cloud storage platform is designed which includes a series of shadow storage services to address these new data management challenges in pervasive computing environments, which called as “SmartBox”.

An efficient way of combining cloud computing and wireless sensor networks [7] is explained. The cloud provides scalable processing power and several kinds of connectable services. This distributed architecture has many similarities with a typical wireless sensor network, where a lot of motes, which are responsible for sensing and local preprocessing, are interconnected with wireless connections. Since wireless sensor networks are limited in their processing power, battery life and communication speed, cloud computing usually offers the opposite, which makes it attractive for long term observations, analysis and use in different kinds of environments.

Several service discovery protocols for wireless sensor networks [9] are proposed. In addition, to reduce power consumption we presented an activation schedule, based on the mapping of the nodes’ operational modes to Bluetooth states. By announcing the activation schedule as a service, are presentation of the state of the nodes is exposed to client applications. The proposed work takes into account of deploying the sensed data in cloud-beez cloud.

3. SOA MODEL
   A. Sensor Network Architecture
   A client application querying the data from sensor network plays the role of service requestor. Sink node acts primarily as a service provider to the external environment. Then sink implements transformation and mapping algorithm to transform the sensor data into a user friendly format for internet accessing. The sensor data is sensed from sensor node which is in the form of quntized xml, assumed to be in compressed format. The sensor node is typically powered with limited capacity. The data which comes from the sensor node is in SOAP format. Inside the SOAP message quantized xml is used. In order to reduce the power consumption the normal SOAP message is optimized into reduced format. This optimized SOAP is sensed from the sensor node to sink node. The sensor data which is sensed from the sensor node is updated on sensor base. This will be useful for performing the future statistical analysis. The client application sends a SOAP request for a specific service to sink by the application running on sun application server. The sink uses the XSLT based transformation and mapping algorithm to transform the data in a suitable format for internet accessing. This is accomplished through web service provided by sun application server.

B. SOA ModelSensorInformation System
   An advanced middleware solution to the problem of integrating a Wireless Sensor Network into the information system of an enterprise at a high abstraction level is proposed through the SOA gateway.
The TOSSIM itself got the packages to simulate real time sensors. Cygwin is used in windows platform to run the TOSSIM. In one cygwin window, the commands as in the Fig 3, is run. As a result a sensor node is simulated and its sensed parameters are written into tossim.txt file in the following path:

C:\ProgramFiles\UCB\cygwin\opt\tinyos-1.x\apps\Sense\tossim.txt. The sense folder also contains two NesC files called configuration (Sense.nc) file and Module(SenseM.nc) file.

XML representation of sensor data: Using XML as a standardized data exchange format in wireless sensor networks is a means to support more complex data management and heterogeneous networks. Moreover, XML is a key feature towards service-oriented sensor networks. Recent work has shown that XML can be compressed to meet the general hardware restrictions of sensor nodes while still supporting updates. Integrating wireless sensor networks in heterogeneous networks is a complex task. A reason is that the absence of a standardized data exchange format that is supported in participating sub networks. Using xml in sensor networks encourages the interchangeability of different types of sensors and systems. The general verbosity of xml conflicts with the limited energy and memory capacities of sensor nodes. For this reason, native xml support has to be based on efficient data binding techniques that saves space, time and energy by eliminating the xml overhead. A good xml data binding solutions for sensor networks has to fulfill the following criteria: Memory efficiency: representing high amount of xml data with a low amount of allocated memory. Runtime efficiency:

using only a minimum number of processing cycles for processing xml data. Processability: allowing to process xmldata
An Efficient and Extensible Service based Sensor Access Architecture

dynamically without an expensive decompressing step. The following code listing specifies the temperature sensor's information:

```xml
<?xml version ="1.0"?>
<sensor>
  <id>01</id>
  <type>sensing</type>
  <parameter>Temperature</parameter>
  <units>56 kelvin</units>
  <date>11</date>
  <month>November</month>
  <year>2009</year>
  <time>11.00</time>
</sensor>
```

**Quantized xml**

Sensor will receive the request from the sink and it will give the sensed information as the response to the intended user. The response which has sent by the sensors has to be taken into account, because the data transmission rate affects the power consumption in the sensor node. So the size of the data has to be reduced. The data capacity reduction will reduce the space utilized, time for data transmission and the energy consumption. This will lead to the increase in the lifetime of the sensor nodes. The quantization is a technique which is used to minimize the data which has to be sent by the sensor nodes. This technique depends on the past transmission records. A comparison is performed on the present data to be sent and the previously transmitted data. Depending on the variations the quantization is performed on the data and then the data is transmitted.

The need of this system is to save time, space and energy, to increase the flexibility, to manage the power consumption and to make the sensor data in the suitable format for internet accessing. The advantages of this system are memory efficiency, runtime efficiency and processability. Memory efficiency is attained by representing high amount of data with a low amount of allocated memory. Runtime efficiency is achieved through minimal number of processing cycles for XML data. And then the power consumption is reduced to a large extent which will increase the lifetime of the sensor nodes. As the data transmitted by the sensor increases the battery power will get lowered. This will affect the lifetime of the sensor. To reduce the data rate the quantization technique is adopted in WSN.

Quantization in the sensor node is the process of converting a large data into a reduced data. The process is initiated by comparing the actual data and the last recently transmitted data.

Let A be the sensor node. The data transmitted from A at the time $t$ is $M$, the size of the data be 4 kb. Let B be the data transmitted at the time $t+1$ from A, whose size is 6 kb. The power consumed for transmitting $M$ be 3 mw and $N$ be 4 mw. As the data rate increases the power consumed also increases. By introducing the quantization method the existing scenario can be changed.

Data transmitted by A at time $t = M$
Data transmitted by A at time $t+1 = N$
where $N > M$
Power consumption of A at time $t = P_1$
Power consumption of A at time $t+1 = P_2$
where $P_2 > P_1$

D. After introducing quantization

**Data rate:**

Data sent by A at time $t = M$
Data has to be sent by A at time $t+1 = N$
Actual data sent by A at time $t+1 = N-M$
where $N' < M$

**E. Power consumption:**

Power consumed to transmit $M$ at time $t = P_1$
An Efficient and Extensible Service based Sensor Access Architecture

Power consumed to transmit M at time t+1 = \( P_2 \)
Actual power consumed to transmit M at time t+1 = \( P_2 - P_1 \)
\[ = P_2' \text{ where } P_2' < P_1 \]

The client sends a request to the sensor node. The request is processed and a quantized response is produced. The quantized response is obtained through the differential value, i.e., the difference between the current data to be sent and the last recently sent data. The quantized response is mapped to a Q-XML template in the server and then the actual response is sent to the client.

F. Sensor as service deployment

```xml
<complexType name="ReadTempSensor">
  <sequence/>
</complexType>

<complexType name="ReadTempSensorResponse">
  <sequence>
    <element name="result" type="string" nillable="true"/>
  </sequence>
</complexType>

<element name="ReadTempSensor" type="tns:ReadTempSensor/>

<element name="ReadTempSensorResponse" type="tns:ReadTempSensorResponse"/>
</schema>
```

G. Sensor registry

A discovery service is a service that enables agents to retrieve Web service-related resource descriptions.

![Image](image)

Figure. 3 sensor service registry

Figure 3 shows the sensor services such as temperature, pressure and flow are available in registry for the clients.

H. SOAP request and response

Here, it is described how the sensed parameters are converted into web service and how they have been deployed. Web services are application components that are designed to support interoperable machine-to-machine interaction over a network. This interoperability is gained through a set of XML-based open standards, such as the Web Services Description Language (WSDL), the Simple Object Access Protocol (SOAP), and Universal Description, Discovery, and
Integration (UDDI). These standards provide a common and interoperable approach for defining, publishing, and using web services.

The lists of services are discovered and invoked by the sensor applications (client), using SOAP messages. The client communications are passed through the Integration Controller. The IC also takes care of the authentication of the users and delivering the required parameters using push interaction pattern. This pattern can be triggered by multitude of events, here an auditable event, trigger (when the process parameters exceeds some threshold) the message sent to the client. J2EE 1.4 SunAppserver is used as service provider. The J2EE 1.4 platform provides comprehensive support for web services through the JAX-RPC 1.1 API, which can be used to develop service endpoints based on SOAP.

**SOAP Request message:**

```
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ReadTempFile xmlns="urn:bul"/>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

**SOAP Response message:**

```
    xmlns:enc="http://schemas.xmlsoap.org/soap/encoding/
    xmlns:ns0="urn:bul"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <env:Body><ns0:ReadTempFileResponse>
        <result>96degreeKelvin 2009-11-14</result>
    </ns0:ReadTempFileResponse>
</env:Body></env:Envelope>done
```

4. **SERVICE ARCHITECTURE**

In this paper, an architecture to integrate the WSN [11] with Internet using the cloud computing technology is proposed. The cloud computing supports the interaction of consumers and WSN through a standard interface.

![Figure 4 Service Architecture](image)

The proposed paper aimed at working on applying and extending the service oriented paradigm to sensor network application engineering such as distributed manufacturing, we derived the requirements for the sharing of sensor networks as new resources in this domain. The necessary abstraction was implemented using the service oriented process parameters, which lead to the intelligence integration into the Internet. This solution has been extended to sensor clouds[ 13], which leads to high availability and hence reliability is achieved. This architecture presents the Integration Controller and Internet can interact using cloud technology. Cloud computing is a way to increase capacity or add capabilities on the fly without investing in new infrastructure, training new personnel, or licensing new software.
The proposed architecture enables users to easily collect, access, process, visualize, archive, share and search large amounts of sensor data from different applications. Supports complete sensor data life cycle from data collection to the backend decision support system. Vast amount of sensor data can be processed, analyzed, and stored using computational and storage resources of the cloud. It allows sharing of sensor resources by different users and applications under flexible usage scenarios. It also enables sensor devices to handle specialized processing tasks. In this architecture the Integration Controller will upload the sensed data to Cloud-beez. Figure 5 shows our implementation of sensor information in Cloud-beez proxy.

**Data Location:** Data should be stored and processed only in specific jurisdictions as define by user. Provider should also make a contractual commitment to obey local privacy requirements on behalf of their customers. Data-centered policies that are generated when a user provides personal or sensitive information, that travels with that information throughout its lifetime to ensure that the information is used only in accordance with the policy[15]

**Backups Of Data:** Data store in database of provider should be redundantly store in multiple physical location. Data that is generated during running of program on instances is all customer data and therefore provider should not perform backups. Control of Administrator on Databases.[15]

Cloud-beez is Elastic Java application platform for EC2. The fastest way for Java EE developers to build, manage and scale applications in the Cloud. Cloud-beez provides easy deployment of test and production environments, a local development model, and strong integration with existing development tools, frameworks, and processes. The IC is responsible to read the sensed data from Sink and deploy into the client end by using Cloud-beez. The following code list is used:

```java
buildXML(strFileName,strCurrentReading,out);
out.println(strCurrentReading);
/* Batch File Running*/
String command = "cmd /C start D:/tomcat50-jwsdp/webapps/servlets-examples/WEB-INF/classes/cloud.bat ";
Runtime rt = Runtime.getRuntime();
rt.exec(command);
} else{
out.println("Error over File making");
```

Figure. 5 sensor service in cloud beez

The Cloud-beez provide separate servers space for our application so any number of users can communicate the servers to get the sensed data. [15]

Data can be redundantly store in multiple physical location. Physical location should be distributed across world. so It can be available everywhere. [15]  

We need Security at following levels  
1. Server access security, 2. Internet access security, 3. Database access security, 4. Data privacy security, 5. Program access Security
5. EXPERIMENTAL AND PERFORMANCE ANALYSIS

The advantages and drawbacks of the proposed combination of wireless sensor networks and SOA environments can be discussed in terms of i) development issues, ii) scalability, iii) interoperability, iv) security, v) business issues. The developments of such systems have the advantage, that several programming languages and different kind of sensor devices can be used and easily interconnected. It provides a lot of flexibility in terms of programming languages and devices, but this comes with the drawback of heterogeneity, which quite often leads to more complexity. The complete analysis summary of test results for implementation using the proposed system is discussed here.

The sensor output is taken from TOSSIM with 100 nodes placed in grid topology with Zigbee communication protocol. The application server played the role of sink. The sensor information from the sink was deployed into the web server in the form of traditional SOAP messages and SOAP with quantized xml messages. The service execution time is compared for performance analysis. The service execution time of the proposed system with traditional SOAP is recorded as in figure 4.

The execution time is almost same when the size of the message is small, i.e., less than or equal to 30k. When the size of the message goes beyond 30k, the execution time of the proposed system shows better performance compared with traditional SOAP based approach. When the message size is more than 30k, the proposed system produces 75-40% better results compared to the traditional SOAP based approach because it used quantized xml as sensor template and also the Service Oriented Approach which provides the necessary asynchronous communication.

![Figure 7 Cloud server throughput](image)

Figure 7 shows the server throughput when the services are deployed in cloud bee兹 proxy. It shows the inherent scalability of the cloud server i.e. it gives the same throughput in terms of number of services execution is same even when the requesting clients are increased.

6. CONCLUSION

In this paper efficient sensor access architecture is proposed which included a Gateway using quantized xml. The lowest hierarchy level is considered to be Wireless Sensor Network. The proposed advanced Gateway implements a stripped down version of the traditional DPWS protocol. It was shown how a WSN that implements the proposed middleware can be integrated into the information system of an enterprise at a
high abstraction level. Moreover, a technique was utilized which compresses and reduces the data volume of XML documents at a level that can be handled by the wireless sensors. The use of the SPWS [16,17] in conjunction with this technique results to savings in the battery life of sensors, in reducing the memory requirements when storing XML exchanged documents and in reducing the traffic with which the sensors load the wireless channel. The architecture proposed which used the quantized xml proved further reduction in memory requirements and was tested with an experiment and proved it is better compared to the existing traditional SOAP based and the SPWS based approach. To facilitate orchestration and aggregation of services into processes and applications, orchestration engine called as sensor service registry is developed that will provide a platform to find the appropriate servers with the services. The SSDL files for sensor services are used and the appropriate service bindings are set to register the services.

This work has taken steps for applying and extending the service-oriented paradigm to sensor network application engineering. Hence the proposed architecture is efficient. Since wireless sensor networks are limited in their processing power, battery life and communication speed, cloud computing usually offers the necessary storage capacity and processing power for long term observations, analysis and use in different kind of environments and projects. This solution has been extended to sensor clouds, which leads to high availability and hence reliability is achieved. The proposed architecture enables users to easily collect, access, process, visualize, archive, share and search large amounts of sensor data from different applications. Thus the sensor information access and monitoring is done in the Cloud environment, which lead to a new paradigm called green Engineering. This leads to the extensible nature of the proposed architecture.

REFERENCES
7. Åke Östmark, Jens Eliasson, Per Lindgren “An Infrastructure for Service Oriented Sensor Networks”,

Published: goplex Publishing


