Integrating Wireless Sensor Networks with Cloud Computing

Khandakar Entenam Unayes Ahmed, Mark A Gregory

School of Electrical and Computer Engineering,
RMIT University, Melbourne, Australia
{khandakar.ahmed, mark.gregory}@rmit.edu.au

Abstract—**Wireless Sensor Networks (WSN) has been a focus for research for several years. WSN enables novel and attractive solutions for information gathering across the spectrum of endeavour including transportation, business, health-care, industrial automation, and environmental monitoring. Despite these advances, the exponentially increasing data extracted from WSN is not getting adequate use due to the lack of expertise, time and money with which the data might be better explored and stored for future use. The next generation of WSN will benefit when sensor data is added to blogs, virtual communities, and social network applications. This transformation of data derived from sensor networks into a valuable resource for information hungry applications will benefit from techniques being developed for the emerging Cloud Computing technologies. Traditional High Performance Computing approaches may be replaced or find a place in data manipulation prior to the data being moved into the Cloud. In this paper, a novel infrastructure is proposed to integrate the Cloud Computing model with WSN.**

I. Introduction

Cloud Computing permits companies to increase capacity quickly without the need for new infrastructure investment and similarly companies can decrease capacity quickly and efficiently. In a recent IBM report it was stated that the "Cloud is a new consumption and delivery model for many IT-based services, in which the user sees only the service, and has no need to know anything about the technology or implementation." [1]. According to the US National Institute of Standards and Technology (NIST) "Cloud Computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resources" [2]. With pools of computing power, network, information and storage resources the cloud offers the use of a collection of services, applications, information and infrastructure. Cloud components can be rapidly orchestrated, provisioned, implemented and decommissioned, and scaled up or down providing for an on-demand utility-like model of allocations and consumption [3]. On-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service are five essential characteristics of Cloud Computing depicted by NIST [4].

Services provided by Cloud Computing can be categorized into three classes: Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS). IaaS provides consumers with an opportunity to consume processing, storage, network, and other fundamental computing resources. Here the consumer is able to store data, deploy and run arbitrary software such as operating systems and applications. The consumer does not need to control and manage the underlying infrastructure but has control over the operating system, applications, storage, and network components. In this service approach, the customer contracts to use servers, the data centre fabric, networking, storage and other facilities [5]. By adopting PaaS consumers can host applications using platforms which include the runtime software necessary to host consumer developed applications. Here also the consumer has no control on the underlying infrastructure but does have control over the deployed applications and application hosting specific configurations. Web 2.0 application runtime, Java runtime, middleware, database, and development tooling are a few example of services provided in this layer [6][7]. In SaaS a vendor supplies hardware infrastructure and software products through a front-end portal. The SaaS concept provides a broad market solution that may include anything from web based email to inventory control and database processing. End users can access the service over the Internet. Service examples include: collaboration, business processes, industry applications, e-Health and CRM/EPR/HR [8].

A vast sea of sensors which have been connected to the global network has started another information revolution and an explosion in our ability to create, store and mine digital gathered information from the sensors. Wireless Sensor Networks (WSN) has moved from an early research topic to the point where the number of implemented WSN is growing rapidly.

Cloud Computing is principally designed and promoted to be data centre centric and efficient interaction with the outside world is an area where improved solutions are being sought. WSN are designed to collect data in the real world, yet, the question arises as to what to do with the data when the organisation that collected the data no longer requires it. There are many reasons for the data to be kept including historical, future research, and re-analysis at some future point in time. There is a possible linkage between WSN and Cloud Computing and the eventual shift of data into the cloud and over time into the public domain.

In this paper an integration framework is proposed between WSN and Cloud Computing. The objective of the integration framework is to facilitate the shift of data from WSN to the Cloud Computing environment so that the scientifically and economically valuable data may be fully utilised.

The paper is organized as follows: Section II presents related work. Section III describes implementation scenarios. Section IV illustrates the proposed framework. Section V presents the conclusion and future work.

II. Related Work

Hassan *et al.* [9] proposed a framework integrating sensor networks and cloud computing. The authors also proposed a new event matching algorithm and showed its performance through experimental analysis. Their proposed framework has a minor flaw in the implementation of the publish/subscribe broker. The paper does not provide a clear understanding of the components and steps that are involved in building the proposed framework. Lee and Hughes [10] proposed an architecture for tangible cloud computing and provided application scenarios along with the tools needed to develop the cloud. The framework proposed is in the initial stages of development and little detail is available. Lim *et al.* [11] proposed integrating a sensor network with grid computing (GRID) but in the current application scenarios GRID may not be appropriate for most purposes. GRID is focused on high performance computing while Cloud Computing concentrates on general purpose applications. This generic nature of the Cloud makes it more attractive and suitable for WSN integration.

III. Application Scenarios

A. Rainforest rehabilitation underpinned by smart CSIRO technology

The CSIRO ICT Centre [12] has deployed wireless sensor nodes in the Mt Springbrook rainforest to measure growing conditions. 185 solar powered nodes were deployed with a total of 640 sensors attached to the nodes. The sensors monitor air temperature, rainfall, soil moisture, wind speed and direction, carbon dioxide concentration, sunshine, cloud cover and fog density. The data collected is growing over time and whilst this data will eventually be no longer of immediate use to CSIRO the future use of this rainforest data at a time of global warming may be considerable and making this data available through the Cloud for a fee would offset the storage and presentation costs.

B. Monitoring water quality and catchment health

In 2008-09 CSIRO implemented the largest land and water-based sensor network to monitor drinking water in Australia. 120 nodes were deployed to record environmental variables such as turbidity and temperature. Nodes co-operated with each other to set up an ad-hoc network to wirelessly transfer data to the nearest base station. The base stations used the third generation mobile phone network to upload data to a remote database. This whole system could be extended one step further by putting the data into the Cloud and thereby permitting future research utilising the data collected [13].

C. Future opportunities

The opportunities identified highlight the potential of using the Cloud as a repository for data collected using WSN. It is also possible to identify that to achieve this outcome a framework including how the data is to be stored, secured and made available is required. Data stored in the Cloud may be stored in a raw format with associated descriptors or manipulated prior to storage.

IV. Proposed Framework

Error! Reference source not found. shows the WSN and Cloud Computing integration framework. The framework components include: Data Processing Unit (DPU), Pub/Sub Broker, Request Subscriber (RS), Identity and Access Management Unit (IAMU), and Data Repository (DR). Data collected from the WSN moves through a gateway to the DPU. The DPU will process the data into a storage format and then send the data to the DR.

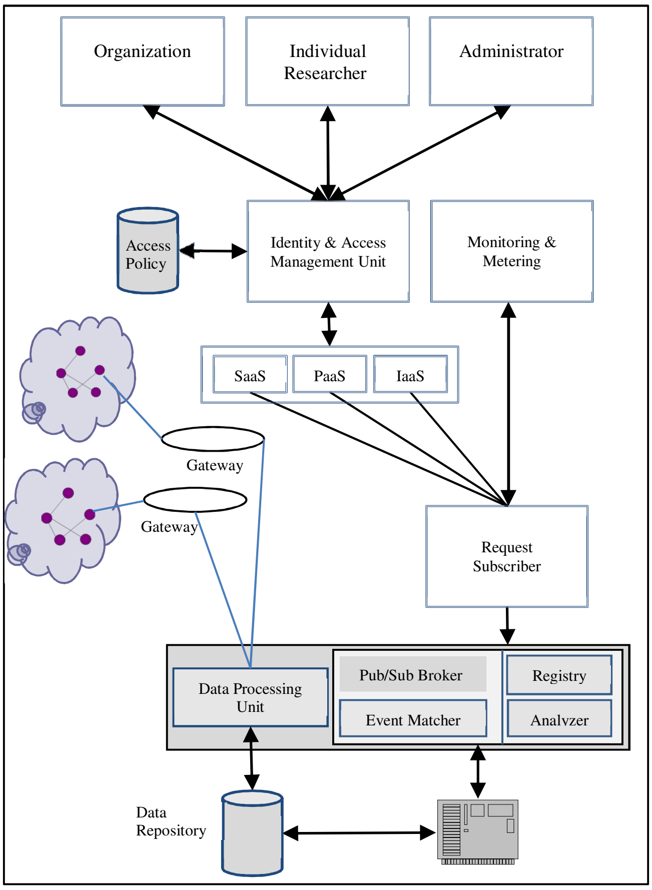

Fig. 1. Sensor-Cloud Integration Framework

Users will connect to the Cloud through the secured IAMU and will be given access on the basis of the policy stored against their user account. After access has been granted users can put forward data access requests. The requests will be forwarded to the RS and the RS will create a subscription on the basis of this request and forward this subscription to the Pub/Sub Broker. Data received in the cloud will be identified by the DPU which will create a published data event and send the event to an event queue at the Pub/Sub Broker. When a new event is published, each subscription is evaluated by the event matcher. Once the event matching process finds a match the published data is made available to the user after further processing is carried out if required.

A. Identity and Access Management Unit:

In this section a prototype Identity and Access Management Unit (IAMU) is described that includes Diffie-Hellman, Kerberos, Role Based Access Control (RBAC) and Extensible Markup Language (XML) based upon previous work [14]. The primary purpose of this model is two-fold: firstly to provide strong authentication between customer and provider; and secondly to provide a strong policy based access control to cloud resources. Clients will communicate with the provider through the Identity and Access Management Unit (IAMU) which will provide both authentication and access control. **Error! Reference source not found.** shows the IAMU system which includes two major components: Access Control Enforcement Unit (ACEU); and Access Control Decision Unit (ACDU). A slight modification has been to the Kerberos authentication implementation [15] [16] introducing a new unit called the Edge Node (EN) which also implements the Diffie-Hellman public key algorithm [17] [18] [19] [20] [21].

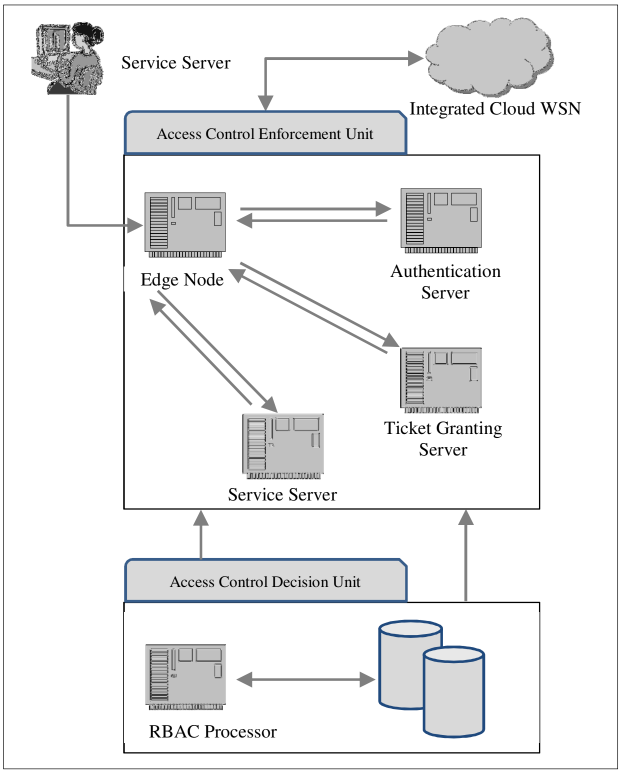

Fig. 2. Schematic Diagram of Overall IAMU System [14]

1) Access Control Enforcement Unit

The ACEU consists of the Edge Node (EN) and three servers: Authentication Server (AS), Ticket Granting Server (TGS), and Service Server (SS). A request arrives at the EN and then it goes to the AS. The EN implements Kerberos to authenticate the client with the AS.

2) Access Control Decision Unit

The ACDU consists of the RBAC Processor and the user policy storage. The ACDU will communicate with the ACEU through the SS.

The authentication process associates users with access policies and after this process has completed the user gains access to data resources within the limitations imposed by the access policies. The proposed framework includes control, group and user management and other information that is stored utilizing XML.

3) Flow of Interaction Between User and IAMU

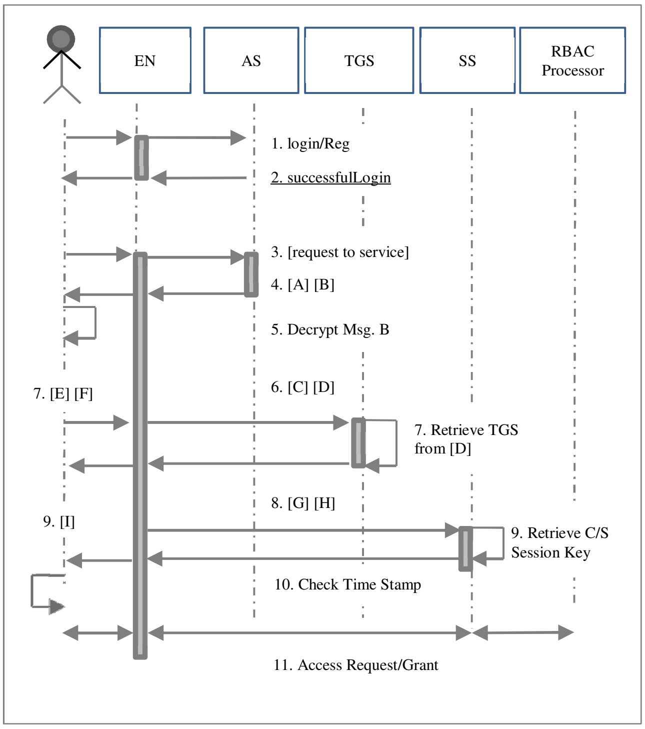

Fig. 3. IAMU Sequence Diagram

Error! Reference source not found. shows the steps involved in the IAMU. The steps are illustrated briefly below [14] [22] [23]:

1. Previous User login or new user registers. This login/register happens securely encrypted by Diffie-Hellman Key.
2. Receives acknowledgement of successful login.

Authentication Steps

3. The Edge Node (EN) sends a [request to service] message to the AS requesting services on behalf of the user.
4. After receiving the service request from a client AS will generate following two messages and send them to client via EN.
   a. Message A: *Client/TGS Session Key* which is encrypted using the public key of the client/user.
   b. Message B: Ticket Granting Ticket (TGT) which includes client ID, client network address, ticket validity period, and Client/TGS

Session Key) encrypted using the secret key of the TGS.

5. After receiving messages A and B from AS, Client will decrypt message A to obtain the Client/TGS Session Key. This session key is used for further communications with TGS.

Client Service Authorization Steps

6. Now, the client will send the following two messages to TGS:
   a. Message C: Composed of the TGT from message B and the ID of the requested service.
   b. Message D: Client ID and the timestamp encrypted using the Client/TGS Session Key (Authenticator).
7. Upon receiving messages C and D, the TGS retrieves TGT out of message C. It decrypts TGT using the TGS secret key. This gives it the Client/TGS Session Key. Using this key, the TGS decrypts message D (Authenticator) and sends the following two messages to the client:
   a. Message E: Client-to-Server ticket (which includes the client ID, client network address, validity period and Client/Server Session Key) encrypted using the SS secret key.
   b. Message F: Client/Server Session Key encrypted with the Client/TGS Session Key.

Client Service Request Steps

8. Upon receiving messages E and F from TGS, the client has enough information to authenticate itself to the SS. The client connects to the SS and sends the following two messages:
   a. Message G: composed the message E received from the previous step (the Client-to-Server ticket, encrypted using the SS secret key).
   b. Message H: a new Authenticator, which includes the client ID, timestamp and is encrypted using Client/Server Session Key.
9. The SS decrypts the ticket using its own secret key to retrieve the Client/Server Session Key. Using the sessions key, SS decrypts the Authenticator and sends the following message to the client to confirm its true identity and willingness to serve the client:
   a. Message I: the timestamp found in client's Authenticator plus 1, encrypted using the Client/Server Session Key.
10. The client decrypts the confirmation using the Client/Server Session Key and checks whether the timestamp is correctly updated. If so, then the client can trust the server and can start issuing service requests to the server.

Granting Access

At this point every service request for a particular resource goes to the SS, and is forwarded to RBAC processor of the ACDU. The ACDU will have access policies (XML) and stored in the database. It should be noted that the ACDU unit database is connected to the AS so that user policies may be obtained. The RBAC processor reads policies and takes decisions accordingly. The RBAC processor decision is forwarded to the SS and on the basis of this decision the SS sends an ACK/NACK to the user station.

B. *Publish/Subscribe Broker*

In publish / subscribe systems, the subscribers provide information requests to the system and publishers submit new information into the system. Upon receiving a publication, the system searches for matching subscriptions and notifies the interested subscribers. This model reduces program complexity and resource consumption [24].

Krishnamurthy [25] provides a framework named TinySIP. TinySIP supports Session Semantics, Publish/Subscribe Semantics, and Instant Messaging. Users usually determine the event supported by a sensor by sending a SIP OPTIONS message. After getting this message the gateway returns a response with a list of supported events. To subscribe to an event a user sends a SIP SUBSCRIBE request message. The TINYSIP framework limits access to specific devices / gateways.

Hall *et al.* [26] proposed the Distance Vector / Dynamic Receiver Partitioning (DV/DRP) publish / subscription model. In DV/DRP, subscriptions flood into the network through intermediate aggregation nodes (IN). The IN forward publications if there are registered requests. However, matching subscriptions to arbitrary data packets is complex and would be difficult to implement.

Souto et al. proposed Mires [27] which is another publish / subscribe communication framework for information generated from a WSN. The communication happens in three phases. In the first phase, the sensor outputs (e.g. temperature and humidity) are advertised by nodes in the network. Using a multi-hop algorithm the advertised messages are routed to the sink node. A user application connected to the sink node can subscribe to the desired data. Finally, subscribe messages are broadcast down to the network nodes. Nodes are able to publish collected data to the network after receiving the subscription request.

In the framework proposed in this paper the pub/sub broker is not directly connected to the gateway. Data collected from sensors will go to DR via a gateway and the Data Processing Unit (DPU). The DPU will trim unnecessary information, format the information into a common storage format and send the data to the DR for storage. Data will have an index which will be stored in registry of the pub/sub broker. The Request Subscriber (RS) will be used to create subscription and the Event Matcher (EM) will find mappings between the subscription requests and published data. Once a mapping is found, the pub/sub broker will start fetching data from the DR and channel the data to the user via cloud's user interface.

C. *Data Processing Unit*

Madden et al. proposed an acquisition query processing system for sensor networks [28]. **Error! Reference source not found.** illustrates the basic architecture followed where queries are submitted, parsed, optimized, and sent into the sensor network, where they are disseminated and processed, with results flowing back up the routing tree that was formed as the queries propagated. Madden et al. developed an adaptive, power-sensitive model for query execution and result collection.

Two Bigtable-alike information storage systems named HBase [29] and Hypertable [30] are built on top of the Hadoop MapReduce [31] programming model. Hypertable allows different logical column to be stored physically together, whilst HBase permits a restricted variation. HBase also supports Bloom filters to improve access speeds. Both of the information storage systems provide similar functionality despite having different architectures. The collected data is organized into tables, rows and columns. Each cell is indexed by a row key, column key and a timestamp. Multiple versions of the same date can be stored using timestamps. An iterator-like interface is available which can be used for scanning through columns. The offered wrappers between both tables and MapReduce [32] encourage the development of novel data processing applications.

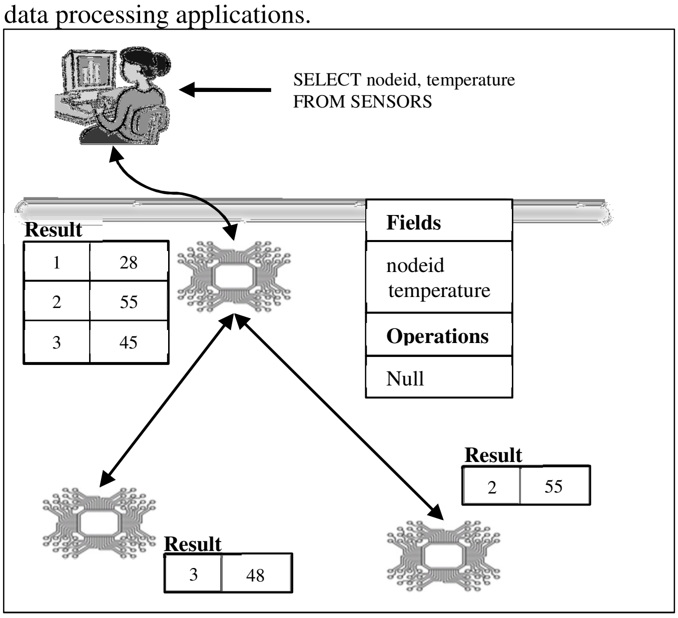

Fig. 4. A Query and Results Propagating Through the Network [26]

The MapReduce programming model would be suitable for distributed database solutions. MapReduce can be used for storing and analysing sensor data and has been used in the development of the proposed WSN Cloud Computing framework.

A database schema has been designed to include a collection of tables for storing individual sensor readings and to provide linkages to related sensor readings. The proposed framework also includes a methodology for analysing and modelling sensor data and various spatial statistics of interest such as sensor location, sensor network type and purpose, collected data and other associated global information. The proposed framework utilises distributed storage principles and the introduction of MapReduce within the data storage regime permits improved data storage and retrieval across the distributed systems.

D. Request Subscriber (RS)

Service requests are created on the basis of user's request to permit access to data stored in the DR or for data gathered from a WSN to be put into the DR. The service requests are passed to the RS unit which will unify the request and send this request to the pub/sub broker to find a mapping with a data index which is stored in the broker registry.

E. Flow of Interaction among Framework Components

The interactions among different components of the proposed framework are shown in **Error! Reference source not found.** including the following steps:

1. The user attempts to login by sending login information.
2. IAMU will authenticate the user and sends ACK if the authentication is successful.
3. After successful login the user will send a service access request.
4. Cloud thread will identify the service type and generate a corresponding request message.
5. Cloud will then send the request message to Request Subscriber (RS).
6. RS will unify the request and create a subscription on the basis of the request received from Cloud thread.
7. Then the RS will send this subscription to the PUB/SUB Broker.
8. DPU will continuously send index of the data to the PUB/SUB broker. This event can happen at any point. PUB/SUB broker will store all of the data indexes in its registry.

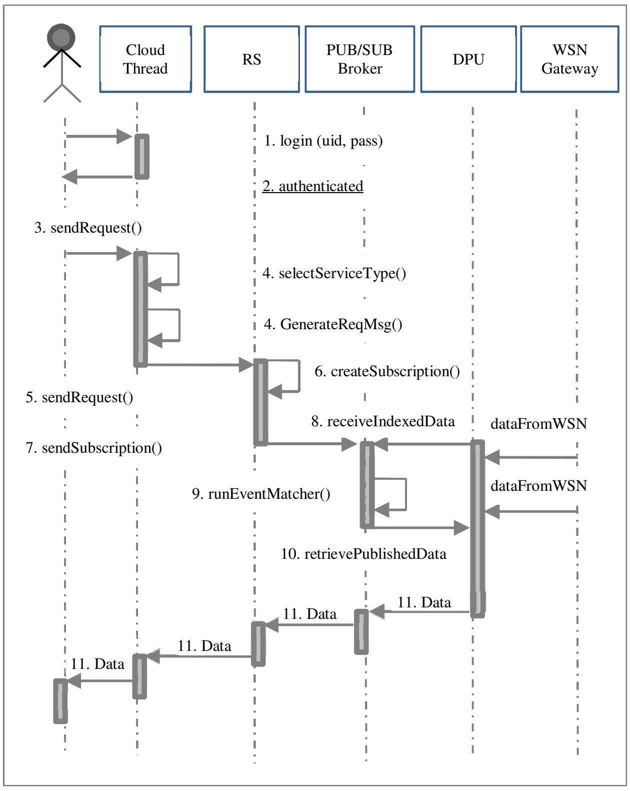

Fig. 5. Sequence Diagram

9. Immediately after receiving a subscription request from RB, Pub/Sub Broker will start EM to find the matched published data for this particular subscription.
10. If Pub/Sub Broker finds any subscription match it will start retrieving data from the DPU.
11. Retrieved data will be forwarded to the user via the RS and the Cloud thread.

V. Conclusion and future work

Integration of WSN and Cloud Computing will provide benefits to organisations and the research community. Organisations will benefit by utilising Cloud storage and an optimised framework for processing, storage and retrieval of WSN generation data. The proposed WSN Cloud Computing framework will provide an optimal approach to user management, access control, storage and retrieval of distributed data. Future work will include further development of the data processing, storage and retrieval methodology. There are parallels to the on-demand video Cloud solutions currently being implemented. Another aspect of future research will be to identify an optimal approach to permit data manipulation prior to publishing.

References

[1] F. Schepers. (2010) Security in Cloud Computing, IBM Tivoli Internet Security Systems. [Online]. Available: http://www.cpdpconferences.org/Resources/Schepers.pdf. Last accessed: 10/11/2010.

[2] P. McDaniel, and S. W. Smith, "Outlook: Cloud Computing with a Chance of Security Challenges and Improvements," IEEE Computer and Reliability Socities 2010, pp. 77-80, Jan. 2010.

[3] R. Marchany. (2010) VA Tech IT Security Cloud Computing Security Issues. [Online]. Available: http://www.security.vt.edu/Downloads/training/Cloud%20Computing%20Security%20Issues.pdf. Last accessed: 2/12/2010.

[4] P. Mell, and T. Grance. (2009) Effectively and Securely Using the Cloud Computing Paradigm (v0.25) NIST. [Online]. Available: http:// csrc.nist.gov/groups/SNS/cloud-computing/index.html. Last Accessed: 10/11/2010.

[5] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, A. Warfield, "Xen and the Art of Virtualization," in *Proc. of 19th ACM symposium on Operating Systems Principles*, Bolton Landing, NY, USA, October 2003, pp. 164-177.

[6] (2010) Google App Engine. [Online]. Available: http://code.google.com/appengine/. Last Accessed: 15/07/2011

[7] (2007) Sales Force. [Online]. Available: http://www.salesforce.com/platform/. Last Accessed: 10/11/2010

[8] A. Dubey, and D. Wagle. (2007) Delivering software as a service - The McKinsey Quarterly. [Online]. Available: http://www.mckinsey.de/downloads/publikation/mck_on_bt/2007/mobt_12_Delivering_Software_as_a_Service.pdf Last Accessed: 15/08/2011

[9] M. Hassan, B. Song and E. Huh, "A Framework of Sensor-Cloud Integration Opportunities and Challenges," in *Proc. of the 3rd International Conference on Ubiquitous Information Management and Communication*, 2009, pp. 618-626.

[10] K. Lee and D. Hughes, "System architecture directions for tangible cloud computing," in *Proc. 1st ACIS International Symposium on Cryptography, and Network Security, Data Mining and Knowledge Discovery, E-Commerce and Its Applications, and Embedded Systems, CDEE 2010*, 23 - 24 October, Qinhuangdao, Hebei, pp. 258-262.

[11] H. B. Lim, Y.M. Teo, P. Mukherjee, V.T. Lam et al., "Sensor Grid: Integration of Wireless Sensor Networks and the Grid," in *Proc. of the 30th IEEE Conf. on Local Computer Networks,* IEEE Computer Society Press, Sydney, Australia, 2005, pp. 91-98

[12] (2010) CSIRO: Wireless Sensor Network: A New Instrument for Observing Our World. [Online]. Available: http://www.csiro.au/science/Sensors-and-network-technologies.html. Last Access: 12/07/2011

[13] (2010) CSIRO: Smart sensors monitoring water quality and catchment health. [Online]. Available: http://www.csiro.au/science/smart-sensors-monitoring-water-quality.html. Last Access: 12/07/2011

[14] K. Ahmed, *Identity and Access Management in Cloud Computing*, 1st ed., LAP Lambert Academic Publishing, Germany, April 2011, ISBN: 978-3-8443-3069-4.

[15] D. Harkins and D. Carrel. (1998) The Internet Key Exchange (IKE), RFC 2409, IETF Network Working Group. [Online]. Available: http://www.ietf.org/rfc/rfc2409.txt. Last Access: 12/01/2011.

[16] F. Ricciardi, MIT Kerberos Consortium. (2007) Kerberos Protocol Tutorial. [Online]. Available: http://www.kerberos.org/software/tutorial.html. Last Access: 22/12/2010.

[17] (2001) SANS Institute InfoSec Reading Room. A Review of the Diffie-Hellman Algorithm and its Use in Secure Internet Protocols. [Online]. Available: http://www.sans.org/reading_room/whitepapers/vpns/review-diffie-hellman-algorithm-secure-internet-protocols_751. Last Access: 12/01/2011.

[18] E. Rescorla. (1999) Diffie-Hellman Key Agreement Method, RFC 2631, IETF Network Working Group. [Online]. Available: http://www.ietf.org/rfc/rfc2631.txt.

[19] (2000) RSA Laboratories, RSA Laboratories' FAQ About Today's Cryptography, Version 4.1, RSA Security Inc., 2000. [Online]. Available: http://www.rsa.com/rsalabs/faq/index.html. Last Access: 12/12/2010.

[20] L. Benjamin. (1997) Diffie-Hellman Method for Key Agreement. [Online]. Available: http://apocalypse.org/pub/u/seven/diffie.html. Last Access: 22/12/2010.

[21] (1993) RSA Laboratories. PKCS #3: Diffie-Hellman Key-Agreement Standard, Version 1.4.Revised November 1, 1993. [Online]. Available: http://www.rsalabs.com/pkcs/pkcs-3/index.html. Last Access: 27/01/2011.

[22] (2011) Wikipedia- Kerberos Protocol. [Online]. Available: http://en.wikipedia.org/wiki/Kerberos_(protocol). Last Access: 5/08/2011

[23] (2011) Kerberos: The Network Authentication Protocol. [Online]. Available: http://web.mit.edu/kerberos/. Last Access: 5/08/2011

[24] D. Tam, R. Azimi, H. Jacobson, „Building Content-Based Publish/Subscribe Systems with Distributed Hash Tables," in *Proc. Of The International Workshop on Database, Information Systems and Peer-to-Peer Computing, September, 2003.*

[25] S. Krishnamurthy, "TinySIP: Providing Seamless Access to Sensorbased Services," *in Proc. of the 1st International Workshop on Advances in Sensor Networks (IWASN)*, 17-21 Jul, 2006, pp. 1-9.

[26] C. P. Hall, A. Carzaniga, J. Rose and A. L. (2006) Wolf A content-based networking protocol for sensor networks. Department of Computer Science, University of Colorado, Technical Report. [Online]. Available: http://www.inf.usi.ch/carzaniga/papers/usi-inf-2006-04.pdf

[27] E. Souto, G. Guimarães, G. Vasconcelos, M. Vieira, N. Rosa, C. Ferraz, J. Kelner "Mires: a publish/subscribe middleware for sensor networks," *Personal and Ubiquitous Computing*, Vol 10, Issue 1 (December 2005), pp. 37 – 44, 2005.

[28] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "TinyDB: An Acquisitional Query Processing System for Sensor Networks", *ACM Transaction on Database Systems*, Vol. 30 No. 1 March 2005.

[29] (2011) HBase. [Online]. Available: http://hadoop.apache.org/hbase/. Last Access: July 2011.

[30] (2011) Hypertable. [Online]. Available: http://www.hypertable.org/. Last Access: July 2011

[31] (2011) Hadoop. [Online] Available: http://hadoop.apache.org/. Last Access: July 2011

[32] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," in *Proc. of the Symposium OSDI*, San Francisco, California, USA, December 2004, pp. 137–150.