



A SCALABLE CLOUD FOR INTERNET OF THINGS IN SMART CITIES

^{#1}TEJENDER SINGH, *Dept of ECE,*

DVR COLLEGE OF ENGINEERING AND TECHNOLOGY (JNTUH), HYD, TS, INDIA.

^{#2}NITHIN RAO.K, *P.hd Scholar, KL UNIVERSITY, VIJAYAWADA,AP, INDIA.*

Abstract: It is crucial to integrate Internet of Things (IoT) with cloud computing to build Smart Cities. However, there are few cloud platforms dedicated for IoT environment in Smart Cities. And existing cloud platforms are complex and difficult to implement with a small scale. Therefore, in this study, we propose a scalable cloud model for Internet of Things (IoT) based on the concept of modularization and service integration. Different from general clouds, the cloud model can be implemented on a small scale in the initial phase, and be extended and integrated in the future. For the resulted cloud platform, we assume that Intelligent City Areas (ICAs), which serve as the basic components of Smart Cities, are the application domains. We also depict the hardware deployment architecture and software implementation model for the cloud platform using Service-Oriented Architecture (SOA) and modular design method. Finally, we demonstrate the effectiveness of our proposed method by designing and deploying a kid safety support system for kindergartens.

Keywords: *Intelligent City Area, Internet of Things, cloud computing, smart cities.*

I. INTRODUCTION

It is crucial to integrate Internet of Things (IoT) with cloud computing to build Smart Cities [1]. IoT is a ubiquitous network, in which various objects are able to interact and cooperate with each other to achieve the instrumented, interconnected and intelligent cities. There are many emerging characteristics of IoT in Smart Cities, for example, heterogeneous data sources, real time processing, environmental dynamism, and spatial-temporal attributes. These characteristics provide great challenges to traditional technologies and cloud computing can be a promising solution to deal with these characteristics. Many well-known companies like Amazon, Google, and Microsoft have offered a variety of cloud computing services. However, existing services focused on online storage and utility computing. There are few cloud platforms dedicated for IoT environment in Smart Cities. Although general cloud computing has some features potential to meet some requirements of IoT, there are many particular needs of IoT such as real-time service delivery and autonomy in subsystems, which are beyond the capability of general clouds. Furthermore, the hardware and software infrastructures of general cloud platforms are quite complex. Although the usage of cloud computing is scalable, to build such a large-scale cloud platform is quite challenging and risky. There are few methods to enable a small-scale implementation of cloud platform to deal with the emerging characteristics of IoT devices in Smart Cities. Therefore, in this paper, we propose a scalable cloud model for IoT in Smart Cities. The proposed cloud model consists of hardware and software components, which are designed based on the concept of modularization and service integration. Modularization is usually characterized by high cohesion around the internal structures and low coupling between each module. It can simplify the cloud structure,

improve performance, and enhance the reliability of processing units as well as the whole cloud [2, 3]. Particularly in this study, we organize the requirements of IoT in Smart Cities into common services, employ Service-Oriented Architecture (SOA) and modular design method, and design a scalable cloud model in accordance the principles of modularity and service integration. Accordingly the resulted cloud platform balances standardization and rationalization with customization and flexibility. In our model, an Intelligent City Area (ICA) is introduced to work as the application domain for our proposed cloud model for IoT [4]. ICA is scalable, extensible and easy to implementation in nature. Client systems which serve as branches of the proposed cloud are deployed in each ICA. Since they are low-coupled, the platform's functions can be easily and economically scaled up. Numerous client systems in ICAs can work together to compose the final cloud platform. The paper is structured as follows. Section 2 reviews related work and highlights research gaps of current research. Section 3 and Section 4 elaborate the hardware deployment architecture and software implement model of our proposed scalable cloud for IoT. Section 5 illustrates a prototype system. Finally, Section 6 summarizes the characteristics of our proposed cloud model and portrays future research directions.

II. RELATED WORK

2.1 Smart Cities and Intelligent City Area

According to Miorandi et al [5], the term 'Smart Cities' is used to denote the cyber-physical ecosystem emerged by deploying advanced communication infrastructure and novel services over city-wide scenarios. IoT technologies are widely deployed in Smart Cities. Intelligent City Areas (ICAs) are the basic components of Smart Cities [4]. An ICA is a region, within 0.5-10 square kilometers, composed



of one or several communities, commercial districts, and urban districts, in which information sensing, sharing, aggregating, analyzing, coordinating technologies are implemented to offer timely services and efficient management for local residents. An ICA normally deploys many advanced ICTs such as IoT, cloud computing, broadband Internet, video surveillance technology, big data management, and information portal service. An ICA can be deemed as a miniature replica of Intelligent Communities and Urban Complex, but it differentiates them in its clear-defined geographical coverage (e.g., School A and Hospital B) and multiple business purposes (e.g., education, food service, security and health care). It is usually small, but extensible. Subsequently a Smart City can be the composite of many ICAs where their infrastructure and resources are under the central planning and management. These features of an ICA make it an ideal application domain for our proposed scalable cloud model.

2.2 Internet of Things

IoT roots in the ubiquitous presence of a variety of objects which are able to interact and cooperate with each other to reach common goals [6]. IoT envisions that digital and physical entities are linked to enable a completely new class of services, not limited to a convergent network [5, 7]. Atzori et al. [6] reviewed different visions of IoT paradigm and related enabling technologies. Gubbi et al. developed an Internet centric IoT architecture [8]. Prior study [8-10] concludes that the increasing number of sensors, diversified sources of information and frequent sampling rate greatly challenge the traditional methods of data collection, storage and analysis.

2.3 Cloud Computing

According to Buyya et al. [9, 10], a Cloud is a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resource(s) based on service-level agreements established through negotiation between the service provider and consumers. Armbrust et al. [11] discussed the top technical and non-technical obstacles and opportunities of cloud computing. Prior research has proposed different models and architectures to enable cloud computing. For example, In [1], a Sea-Cloud-based Data Management mechanism to manage and query massive sensor sampling data is proposed. In [12] a cloud based conceptual architecture for IoT is proposed. In [13], the sensing as a service on cloud is illustrated. In [14], a conceptual framework for data procured from distributed devices is presented. In [15], a process and a system architecture are proposed that enables developers and business process designers to dynamically query, select, and use running instances of real-world services. Table 1 compares architectures of typical existing cloud models.

2.4 Relation between Cloud Computing and Internet of Things

Prior study has identified the needs for IoT-Cloud convergence. Firstly, IoT systems need to connect enormous heterogeneous sensors which continuously gather the status data of real-world objects and produce a large volume of information [1, 13]. IoT can directly benefit from the immense storage capacities of cloud computing infrastructures [14]. Secondly, IoT applications that interact with physical devices need high-performance computing power to enable real time processing of mass data [13]. More specifically, not only distributed storage is necessary, but also central processing capacity is required [14]. Distributed cloud computing can provide powerful utility to perform complex computing [7, 14]. Thirdly, IoT applications are deployed in dynamic environments, which is difficult to provide sufficient computational resources. As a model for on-demand services, cloud computing can access to a shared pool of virtualized resources that can be dynamically reconfigured [10].

III.HARDWARE DEPLOYMENT ARCHITECTURE

We first elaborate the hardware architecture of the proposed scalable cloud model for IoT in Smart Cities. The hardware deployment architecture is based on our previously proposed architecture of ICA [4], the general architecture of cloud platform [9] and the recommendations of the International Telecommunication Union for IoT [32]. The architecture considers the hierarchy, time sequence, and logical order of service flows and data processing. The architecture is partitioned into five layers depicted in Fig. 1: the information acquisition layer, the data storage layer, the communication transmission layer, the application service layer, and the service delivery control layer.

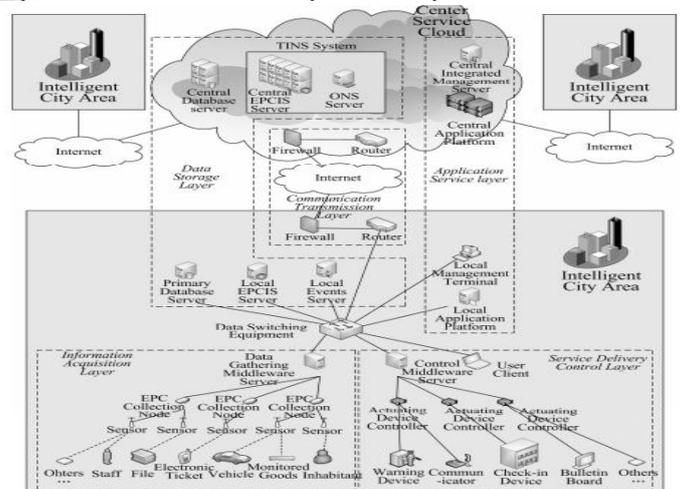


Fig. 1. Hardware deployment architecture of the scalable cloud for internet of things

3.1 Information Acquisition Layer



The information acquisition layer, which serving as the foundation of the architecture, is mainly responsible for data gathering and data transfer to native data storage nodes in ICAs. It is composed of RFID (Radio Frequency Identification) tags, sensor cluster, signal transmission line, and data gathering middleware server. As one of the technological pillars of Internet of Things, RFID is an automatic non-contacting identification technology, which can automatically recognize certain targets by radio frequency signal and access to relevant data [5]. In the conception of IoT, specification and interoperability information stores in RFID tags. RFID tags can work in a variety of harsh environments [7, 33]. There are three types of RFID tags: passive, semi-active, and active. For the initial construction phase of ICA, we suggest to reuse existing local transmission networks for newly-built IoT infrastructure. Wireless data communication network such as Wireless Fidelity (Wi-Fi) network is preferred, when additional transmission infrastructure is necessary. During the process of information perception, data gathering middleware coordinates communication between sensing devices and transport nodes, which adopt different interface standards. After preprocessing and converting of data gathering middleware server, data collected by Electronic Product Code (EPC) collection nodes are transmitted to local events server.

3.2 Data Storage Layer

The data storage layer contains two parts: the local data storage management system and the central data storage management system. a. Local data storage management system As a branch of the cloud center, local data storage management system serves an ICA where it is located. It consists of three parts: the local EPC Information Service (EPCIS) server, local events server, and local primary database server. All EPC information of service entity objects are managed by an ICA are stored in each local EPCIS server. A local events server records the information of RFID tag transmitted by data gathering middleware server at a regular intervals. It works in slow-sync mode and event information is uploaded to cloud periodically. Finally, a local primary database server stores various data to provide service for local application platform, including sensor information, actuating devices information, RFID tagged entity information, user information, and direct access data from IoT applications. b. Central data storage management system A central data storage management system provides entire ICA cluster integrated services. It is located in a center service cloud, which comprises Tag Information and Name Service System (TINS) and central database servers. A TINS has two components: the central EPCIS server and the Object Name Service (ONS) server. The central EPCIS server stores all the unique identifying information and specified part of the detailed information of

service objects in ICA, such as goods, inhabitant, and other tagged entities. Data in the central EPCIS server is a dynamic subset of data in local EPCIS servers. Additionally, it can serve as a disaster recovery system of local EPCIS servers. The ONS Server stores the entire service object information including identification and Uniform Resource Locators (URLs) to process information access request. The static ONS server is usually adopted in the process of practical implementation. The central database server provides data services for multiple applications in different ICAs. It is also in charge of multi-stream data fusion including integrated storage and management of users, infrastructures, and other information belonged to different ICAs. In addition, it backups important data stored in local databases including data of selective events, ICA information, user information, and application service information, for disaster recovery purposes.

3.3 Communication Transmission Layer

The data transmission between a cloud and local ICA systems is accomplished in communication transmission layer. In the context of Internet of Things, public network is still the main carrying network, including Internet and mobile communication network. Local existing telephone network, CATV network, and optical fiber network are all feasible for communication carriers. In this layer, interfaces between the center server and local servers are dynamic, depending on the transmission strategy corresponding to different characteristics of each ICA.

3.4 Application Service Layer

The application service layer contains four components: the central application platform, the central integrated management server, the local application server, and the local management terminal. The central application platform employs the Platform as a Service (PaaS) model. It offers an integrated application development and operating environment to third-party application developers. All operation systems, databases, and module interfaces conform to the same set of standards. These standards, including protocols and APIs, make sure existing resources function efficiently and sufficiently. These standards also guarantee the comparability of applications running on different ICA service platforms. The central application platform also serves administrators of local ICAs and end-users, such as local residents and external users. The central application platform can offer local administration services, such as application release, data backup, software update, and system patching, by remote centralized configuration. The platform can agilely invoke and assemble application modules to offer local users customized services, such as identity authentication, remote monitoring. The platform can also provide information access and processing services to authorized internal and external users, such as dataset access and data mining services. Supported by the central database



server, the central integrated management server supports for the overall management of user registration, authorization, access control, function customization, certification, and the other global services. Besides, it maintains the function module library, software library, and service application catalog and provides personalized services to local ICAs. The local application server is deployed in each ICA. It offers IoT applications runtime environments. After receiving instructions from central integrated management server, a local application downloads corresponding service modules and software from the central application platform, assembles them, and then executes to provide services. . As a branch of the central integrated management server, the local management terminal provides local administration and user access points. The local management terminal can reduce the workload of the cloud.

3.5 Service Delivery Control Layer

The service delivery control layer comprises a control middleware server, actuating devices, and user clients. The control middleware server receives instructions from both the central and local application platforms, then sends transformed operational orders to the actuating devices by actuating devices controller. When tasks are complex, Programmable Logic Controller (PLC) is adopted. Actuating devices, such as warning devices, checkin devices, and bulletin boards, can perform service delivery. Meanwhile, user clients can deliver data-relevant services supported by the center application platform.

IV. SOFTWARE IMPLEMENTATION MODEL

Another important element of our proposed scalable cloud model for IoT is the software implementation model. In this study, we propose the software implementation model based on the multicore client cloud computing model [34] and the SaaS model [11]. The multicore client cloud computing model purports that clients serve as extensions of a cloud which just like the multiple processors on a single chip in general multiple core architecture. When processing responsive tasks which require less computing resources, these tasks will be distributed to the clients. The cloud maintains information encrypted on servers with client’s private key to guarantee information privacy. The cloud can also redistribute these tasks as needed to keep the elasticity. The cloud can automatic allocate server resources to keep capacities of computation, storage and communication available on demand. The clients are physically located in different ICAs and the cloud can start with small-scale implementation. SaaS is the core of our proposed cloud, where web services can be accessed anywhere, anytime, by any device [35]. Consequently, our cloud model adopts web service to offer layered service delivery. According to the

common types of IoT services, different IoT sub-services are classified to facilitate the services integration, and grouped into several modules. Fig. 2 illustrates our proposed software implementation model. The model leverages on the following principles: modularity, convenience in the initial construction, unity, and extensibility.

4.1 Application Delivery Node Implementation Method

The application delivery node is a software system which provides deliverables to users. It is composed of a realtime service system and a data service client. The real-time service system responds to user’s interaction in real time, such as Electronic Toll Collection (ETC), security alarm, automatic ticketing and so on. The system comprises four parts: the element information collecting software, the production database, the real-time service client, and the data extraction and conversion agent. The element information collecting software accomplishes the acquisition, transmission, and storage of information gathered by IoT devices such as RFID reader, video monitor, and so forth. After preprocessed, the data is sent to the production database, which is generally large-scale with fine granularity. Necessary data for the realtime service client to deliver service, such as entity basic information and permission information, are also stored in the production database. The data extraction and conversion agent processes data according to requests from the cloud, and then transmits data to the cloud in a way of limited controlled access, such as web service, rather than direct link to the database. The data service client offers data analysis services provided by the cloud to internal and external users. It can be customized according to customer needs and accessed by thin clients such as web browsers if necessary.

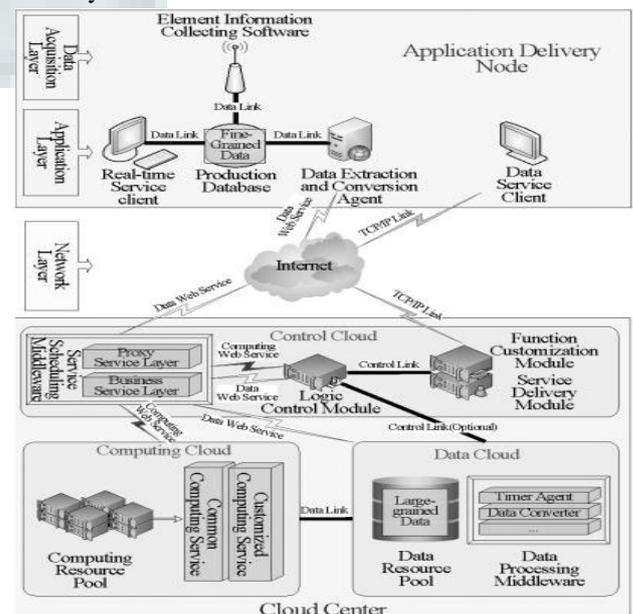


Fig. 2. Software implementation model of the cloud platform for Internet of Things

4.2 Cloud Center Implementation Method



The cloud center is composed of three parts: the data cloud, the computing cloud, and the control cloud. The data cloud and the computing cloud are usually collocated to reduce the cost and improve the efficiency. The control cloud can access the computing cloud and the data cloud by web services rather than direct data links. So the control cloud and the other two clouds can be geographically separated.

a. Data Cloud

The data cloud consists of a data resource pool and a data processing middleware. Coarse granular data which are processed by data processing middleware is stored in data resource pool. The data are utilized for analysis and prediction. The data processing middleware is in charge of data integration, cleansing, and conversion. For instance, a timer agent extracts data from the application delivery node periodically and the data converter transforms the data to a specific format. And other data processing related modules can be deployed in data cloud. The data cloud serves as the data provider for both real-time services and data analysis services for IoT.

b. Computing Cloud

The computing cloud is in charge of the computing functions of the cloud center. Common computing services and customized computing services are provided by the computing resource pool which is hosted by high performance server clusters. Computation capacity can be packaged to form common computing services. For example, the analysis service can be classified into three types: descriptive, predictive, and suggested services. The descriptive service includes crystal report, dashboard, and balanced scorecard; The predictive service includes data mining and statistical prediction. And the suggested service includes optimization, simulation, and decision model. Furthermore, the computing cloud also provides customized computing services such as the preprocessing service of multimedia content. The computing cloud is connected to the data cloud by data links directly to get efficient access. Then data-related functions are released to the control cloud as web services.

c. Control Cloud

Serving as the kernel of the cloud center, the control cloud is responsible for the service logic processing and web services management ranging from receiving user requests to delivering services. Generally speaking, the control cloud includes four parts: the service scheduling middleware, the logic control module, the function customization module, and the service delivery module. The service scheduling middleware such as Enterprise Service Bus (ESB) is used to define, release, and invoke web services. It can be divided into two layers: the business service layer and the proxy service layer. The business service layer includes basic business services, which are supplied by computing cloud directly. The proxy service layer invokes, sorts, assembles,

and re-constructs basic business services according to specific business processes which are predefined. In other words, the service scheduling middleware in control cloud reassemble common storage and computing resources provided by the other two clouds to form new dedicated services according the characteristics of IoT business processes. And these dedicated services are provided for logic control module via several kinds of web services such as data web service, computing web service and so forth. The function customization module process the requests from application delivery node. A user customizes data, computing resources, invocation method, processing flow, and so forth. The function customization module interpret the requests and reorganize them into specific format for the logic control modules. The logic control module builds workflows and invokes all the necessary resources in the cloud, calling corresponding services deployed in the service scheduling middleware. As services in service scheduling middleware are predefined, deployed and available to process the incoming workflows, the logic control module can get the processing results, including result content and presentation structure. And the logic control module transfers them to the service delivery module. The service delivery module processes the results according to the presentation structure, and then delivers to users through data service clients. In order to accept user requests and deliver processing results, the logic control module establishes a control connection to the function customization module and the service delivery module. The control link can be established between the logic control module and the data cloud to facilitate data access if network conditions permit.

4.3 Interaction between Cloud Center and Application Delivery Node

The cloud center interacts with the application delivery node by two methods: the direct linkage by TCP/IP protocol and the controlled access by web services. a. TCP/IP The direct linkage by TCP/IP protocol is mainly used when user access the control cloud through thin clients, such as web browsers, to issue service requests and accept service responses. The method of such direct linkage aims to provide end users an easy, fast, and effective way to get services. b. Web Service Web service is normally used in the process of the application delivery node to provide data resource to the cloud center, in order to ensure reliability, customizability, and controllability of the data transmission. In other words, the administrator of the application delivery node can offer partial access to data for the cloud center for external users, which ensures the autonomy, independence, and security of the node system. In this way, the cloud center backups the predefined data which is difficult to obtain again, serving as a disaster recovery center of the application delivery node.



V. AN ILLUSTRATED PROTOTYPE SYSTEM

According to the hardware deployment architecture and software implementation model of our scalable cloud model for Internet of Things, a prototype system is designed and implemented in the context of kindergarten safety supports. The prototype system is a distributed cloud platform named Intelligent City with Internet of Things (ICWIOT) service platform. The platform implement varies types of functions by different sub-platforms and modules. The sub-platforms and modules are not only logically independent, but also can be geographically separate. The whole platform is composed of four parts depicted in Fig. 3, i.e., the Kid Safety Support System (K3S), the ICWIOT data cloud, the ICWIOT computing cloud, and the ICWIOT control cloud.

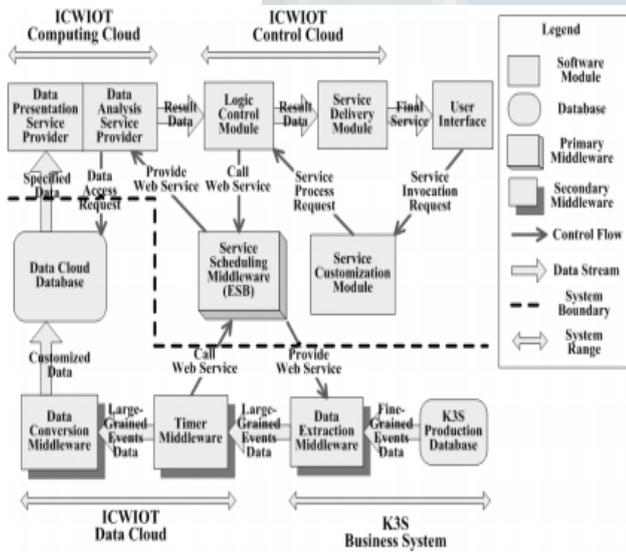


Fig. 3. Prototype system work flow

VI. CONCLUSION AND FUTURE WORK

Cloud platforms for IoT play a vital role for Smart Cities. This paper elaborates a scalable cloud model for Internet of Things based on the concept of modularization and service integration. Different from general clouds, the cloud model can be implemented on a small scale in the initial phase, and be extended and integrated in the future. For the resulted cloud platform, we assume that Intelligent City Areas (ICAs), which serve as the basic components of Smart Cities, are the application domains. We also depict the hardware deployment architecture and software implementation model for the cloud using Service-Oriented Architecture (SOA) and modular design method. Finally, we demonstrate the effectiveness of our proposed method by designing and deploying a kid safety support system for kindergartens. However, there are several limitations of the study. The prototype system is only experienced a small-scale implementation, and the performance of our cloud model is not fully evaluated. Since the proposed scalable cloud model focuses on services in IoT rather than general

computation and storage, traditional metrics may not appropriate to evaluate the performance of our cloud model. Therefore, in the future we will improve our prototype system and expand the implementation scale, and we will continue our research on performance evaluation of different cloud models. We are also interested in algorithms for service resource scheduling and optimization to improve the adaptability and operating efficiency of the service platform. Notwithstanding these limitations, the study has important implications. The design of our scalable cloud model can successfully response the challenges of IoT in Smart Cities. The proposed architecture and implementation model provide a feasible solution to build a cloud platform for IoT, which supports the step-by-step construction of Smart Cities.

REFERENCES

- [1] Z. M. Ding, J. J. Xu, Q. Yang, "SeaCloudDM: a database cluster framework for managing and querying massive heterogeneous sensor sampling data," *Journal of Supercomputing*, Vol. 66, pp. 1260-1284, Dec 2013.
- [2] P. Tonella, "Concept analysis for module restructuring," *Ieee Transactions on Software Engineering*, Vol. 27, pp. 351- 363, Apr 2001.
- [3] K. Praditwong, M. Harman, X. Yao, "Software module clustering as a Multi-objective search problem," *IEEE Transactions on Software Engineering*, Vol. 37, pp. 264-282, Mar-Apr 2011.
- [4] Y. Fu, S.-L. Jia, "A study on intelligent city area based on Internet of Things and cloud computing," *Microcomputer Information*, Vol. 27, pp. 71-73, 2011.
- [5] D. Miorandi, S. Sicari, F. De Pellegrini, I. Chlamtac, "Internet of Things: Vision, applications and research challenges," *Ad Hoc Networks*, Vol. 10, pp. 1497-1516, Sep 2012.
- [6] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, Vol. 54, pp. 2787-2805, Oct 2010.
- [7] P. Liu, "A brief research of the Internet of Things," in *Recent Trends in Materials and Mechanical Engineering Materials, Mechatronics and Automation*, Pts 1-3. Vol. 55-57, Q. Luo, Ed., ed, 2011, pp. 1679-1682.
- [8] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, Vol. 29, pp. 1645-1660, Sep 2013.
- [9] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, Vol. 25, pp. 599- 616, Jun 2009.
- [10] L. M. Vaquero, L. Rodero-Merino, J. Caceres, M. Lindner, "A break in the clouds: Towards a cloud



definition," *Acm Sigcomm Computer Communication Review*, Vol. 39, pp. 50-55, Jan 2009.

[11] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, et al., "A view of cloud computing," *Communications of the ACM*, Vol. 53, pp. 50-58, Apr 2010.

[12] L. Liu, X. Liu, X. Li, "Cloud-based service composition architecture for Internet of Things," *Internet of Things*, Vol. 312, pp. 559-564, 2012 2012.

[13] B. B. P. Rao, P. Saluja, N. Sharma, A. Mittal, S. V. Sharma, "Cloud computing for Internet of Things & sensing based applications," in *2012 Sixth International Conference on Sensing Technology*, ed, 2012, pp. 374-380.

[14] G. Suci, A. Vulpe, S. Halunga, O. Fratu, G. Todoran, V. Suci, *Smart Cities Built on Resilient Cloud Computing and Secure Internet of Things*, 2013.

[15] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, D. Savio, "Interacting with the SOA-based Internet of Things: Discovery, query, selection, and on-demand provisioning of web services," *IEEE Transactions on Services Computing*, Vol. 3, pp. 223-235, Jul-Sep 2010.

