TOWARDS PERVASIVE COMPUTING ENVIRONMENTS WITH CLOUD SERVICES

Ahmed Youssef

CCIS, King Saud University, Riyadh, KSA
Faculty of Engineering, Helwan University, Cairo, Egypt
ahyoussef@ksu.edu.sa

ABSTRACT

Pervasive computing aims at creating environments saturated with embedded and portable computing devices surrounding the users and providing them with many interesting services. It has been gradually applied into our everyday life and its applications are becoming more and more extensive day by day. However, exploiting its full potential is difficult due to some inherent problems such as limited scalability of users and devices, limited availability of software applications, resources scarceness in embedded gadgets, and frequent disconnection and finite energy of mobile devices. Consequently, a wide range of applications such as image processing, multimedia, and social networking are difficult to run in pervasive computing environments. Cloud Computing (CC) platforms possess the ability to overcome these discrepancies with their scalable, highly available and resource pooling computing capabilities. In this paper, we examine how interesting characteristics of CC can contribute to solve some inherent problems in pervasive computing environments. We propose an infrastructure and a protocol that incorporate pervasive computing environments with cloud services and show that by embedding the power of CC services into pervasive computing environments we can significantly enhance services provided by these environments.

KEYWORDS

Pervasive Computing, Ubiquitous Computing, Mobile Computing, Cloud Computing.

1. INTRODUCTION

Pervasive computing, also called ubiquitous computing, aims at creating environments saturated with all kinds of embedded and portable computing devices that communicate via wireless technologies with the users as they move within these environments. In such environments, the users are surrounded by many devices such as PDAs, mobile phones, and wireless sensors that provide different services to them. Pervasive computing applications have many different characteristics that distinguish them from traditional computer applications. One important characteristic is that the embedded computing and communication capabilities are *invisible* to users (i.e., users are unaware of the computing embedded devices). In such environments, information pursues the user rather than the user pursues information. Another prominent feature is the application of *wireless sensors* that are used to implement certain specific tasks. *Mobility* of users is an added characteristic to pervasive computing environments. Pervasive applications are always running and are available to users anywhere and at anytime. Furthermore, They are able to adapt to the environments when users with mobile devices move from one place to another (i.e., reconfigurable to available services). In fact, the trend of pervasive computing encompasses many advanced technologies such as Wireless Sensor Networks (WSN), Distributed Processing, and Mobile Computing [1,2].

Pervasive computing is expected to create a revolution by providing services that use the knowledge of surrounding physical places to meet user intention and preferences [3]. However,

DOI : 10.5121/ijasuc.2013.4301

exploiting its full potential is difficult due to some inherent problems such as *limited scalability* of users and devices, *limited availability* of software applications and information, *resources scarceness* in embedded gadgets and *frequent disconnection* and *finite energy* of mobile devices. As a result, a wide range of applications are difficult to run in pervasive computing environments. These applications fall into different areas such as image processing and recognition, text translation, speech recognition, multimedia, social networking, and sensor data applications. Cloud Computing (CC) platforms possess the ability to overcome these discrepancies with their scalable, highly available and resource pooling computing resources. CC platform is an aggregation of hardware, systems software and application software where the application software is delivered as a service over the Internet, Software-as-a-Service, and the systems software and the hardware in data centres provide Platforms-as-a-Service, and Infrastructure-as-a-Service. The concept behind CC is to offload data and computation to a remote resource provider (the Cloud or the Internet) which offers many interesting characteristics such as: on-demand self-service, broad network access, resource pooling, rapid elasticity, multitenancy, and scalability [4].

The concept of offloading data and computation in clouds, is used to address the inherent problems in pervasive computing by using resource providers (i.e., cloud resources) other than the embedded devices themselves to host the execution of user applications and store users' data. The problems are addressed as follows: 1) by exploiting the computing and storage capabilities (resource pooling) of the cloud, pervasive intensive applications can be executed on low resource and limited energy pervasive devices, 2) the broad network access of the cloud overcomes the limited availability and frequent disconnection problems since cloud resources are available anywhere and at anytime, 3) the infrastructure of cloud computing is very scalable, cloud providers can add new nodes and servers to cloud with minor modifications to cloud infrastructure, therefore; more services can be added to the cloud, this allows more mobile users to be served and more embedded and portable devices to be connected. In the paper, we investigate how interesting characteristics of CC can solve some inherent problems in pervasive computing environments and show that by embedding the power of CC computing resources and services into pervasive computing environments we can significantly enhance services provided by these environments. The rest of this paper is organized as follows: in sections 2 and 3, we review pervasive computing and cloud computing respectively, in section 4, we introduce a proposed infrastructure for the computing environment that empower pervasive computing with cloud services (UbiCloud) and suggest a protocol that organizes interaction among entities in this environment, in section 5, we give our conclusions and future work.

2. PERVASIVE COMPUTING

In 1991 Mark Weiser described his vision of ubiquities (pervasive) computing as a computing environment saturated with computing and communication capabilities, yet gracefully integrated with human users [5]. In such environments, the portable and embedded devices are completely connected and do not require any preload infrastructure. The traditional wire interconnection between nodes are fully replaced by wireless transmission protocols and as a results the nodes can move freely inside an environment or across different environments [2]. Therefore, Wireless Ad-hoc Networks (WAN) and Mobile Ad-hoc Networks (MANET) are good representations of pervasive computing.

However, the research agenda of pervasive computing subsumes that of wireless and mobile computing and goes much further as illustrated in Fig.1. In this figure, as one moves from left to right, new problems are encountered. In addition, the solution of many previously encountered problems became more complex. Specifically, pervasive computing adds four research problems to those encountered in distributed and mobile computing. The first research problem is the *creation of smart spaces*, by embedding computing infrastructure in building infrastructure a smart space is created. Examples of these smart spaces are: the automatic adjustment of lightening, cooling, and heating levels in a room based on occupant's e-profile. The second research problem is *invisibility and context awareness*, which means the complete disappearance of embedded devices from the user's consciousness. This implies that pervasive computing environments must seamlessly tailor itself to meet user expectations and intention. The third research problem is the *localized scalability and adaptability*, scalability in pervasive computing differs from its broadest sense in that it is related to physical distances. The density of interactions has to fall off as the user moves away from the smart space; otherwise, both the user and his computing system will be overwhelmed by distance interactions that are of little relevance. The fourth research problem is the development of techniques for *masking uneven conditioning* of environments, which means achieving uniform dissemination of pervasive computing technology into different infrastructures. With uneven conditioning, there will be huge differences in the smartness of different environments based on nontechnical factor such as organizational structure, economics, and business model. This large dynamic range of smartness can nullify invisibility of pervasive computing technology [1].

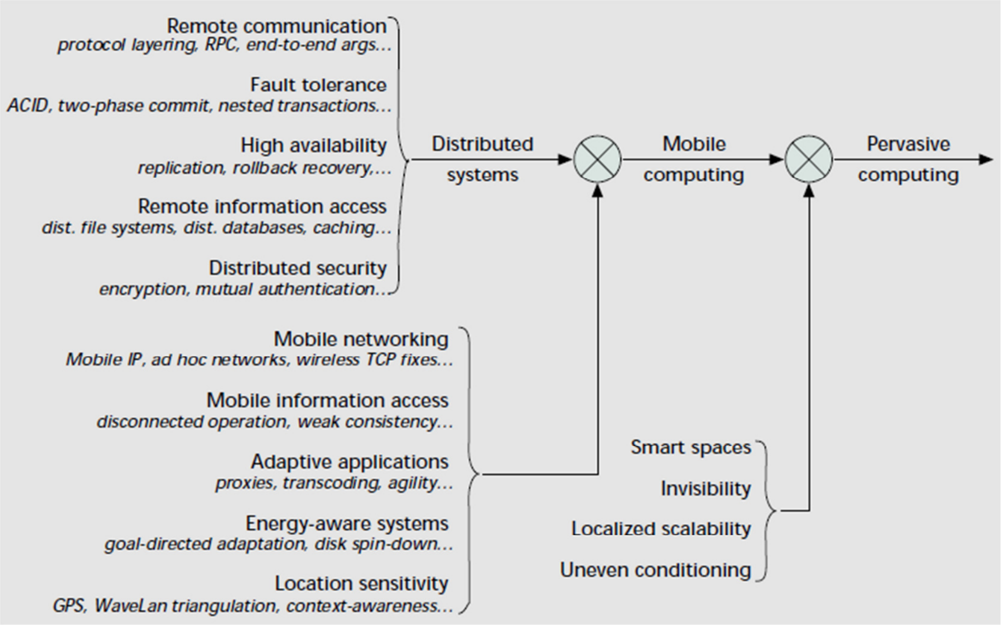
Remote communication
protocol layering, RPC, end-to-end args...
Fault tolerance
ACID, two-phase commit, nested transactions...
High availability
replication, rollback recovery,...
Remote information access
dist. file systems, dist. databases, caching...
Distributed security
encryption, mutual authentication...
Distributed systems
Mobile computing
Pervasive computing
Mobile networking
Mobile IP, ad hoc networks, wireless TCP fixes...
Mobile information access
disconnected operation, weak consistency...
Adaptive applications
proxies, transcoding, agility...
Energy-aware systems
goal-directed adaptation, disk spin-down...
Location sensitivity
GPS, WaveLan triangulation, context-awareness...
Smart spaces
Invisibility
Localized scalability
Uneven conditioning

Figure 1. Research problems in distributed, mobile, and pervasive computing [1].

In order to better understand the characteristics of pervasive computing environments, Kevin Li [6] gave the following example scenarios:

Scenario 1: Bill opens his refrigerator to take out a beer, and he notices that there is only one bottle left. Bill then scans the UPC of the beer with the scanner attached to the refrigerator. Bill makes a note on the refrigerator device that he needs to purchase beer by Wednesday. Bill's virtual shopping list is maintained by his refrigerator, when he scanned the beer, the refrigerator adds beer to the shopping list and notifies his car about the need for beer. The next day, on his way home from work, his car constantly monitored the available services; when his car detected a hypermarket service, it notified him that he is near the BigMart hypermarket and he should stop by to buy the beer. However, Bill did not act to the suggestion. Wednesday rolls around and he still have not bought the drinks, a notification is sent to his mobile phone by the refrigerator to alert him that he should by the beer by Wednesday. When it's Wednesday and the refrigerator has not received a "beer purchased" notification, it sent another notification to his mobile phone.

Scenario 2: It is 5:00 pm and John's work day is just ending. As he is getting ready to leave the office, his phone rings. It is Sara asking him to meet her at the local shopping mall. John agrees to meet her and notifies his mobile phone about the appointment. His mobile phone is able to connect to his telecom provider WLAN and fetch the directions through Google maps. In the car, the GPS of his mobile phone displays directions to the Mall. While on-route, John feels traffic jam and would like to get to his destination via some quicker route. He instructs his mobile device, which can now connect to cars around him, to ask these if they are aware of a faster route. The device contacts its neighbors and returns with an alternate route's directions. John takes the different roads and arrives at the mall's entrance 10 minutes earlier. He decides to use the extra time by checking out local stores for a bargain on some present for Sara. His mobile device finds out from the mall directory server stores that sell such gifts, contacts them to find out possible presents within a $50 and suggests ideas to John. Upon Sara's arrival, John asks his mobile device to suggest available restaurants (the device cached such information during the mall exploration) and lets Sara pick one. She chose the closest Indian restaurant that indicates it has an available table with no waiting period.

3. Cloud Computing

Cloud Computing (CC) introduces a new Internet-based environment for on-demand, dynamic provision of reconfigurable computing resources [10-14]. The definition of CC provided by the National Institute of Standards and Technology (NIST) [7,15] states that: "*Cloud computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage) that can be rapidly provisioned and released with minimal management effort or service provider interaction*". Figure 2 shows the NIST visual model for cloud computing. NIST defines four Cloud deployment models (*public*, *private*, *hybrid* and *community*) which describe the scope of services offered to cloud customers. Private cloud's infrastructure is owned and used by a single organization. It can be managed by the organization itself or by a third party internally or externally. In community cloud, infrastructure is shared between many organizations with common concerns such as security, policy, mission, and compliance. These clouds are managed and hosted internally or externally through a third party. Hybrid cloud is a combination of two or more cloud infrastructures that remain unique entities, but are bounded together to provide advantages of multiple cloud structure [8].

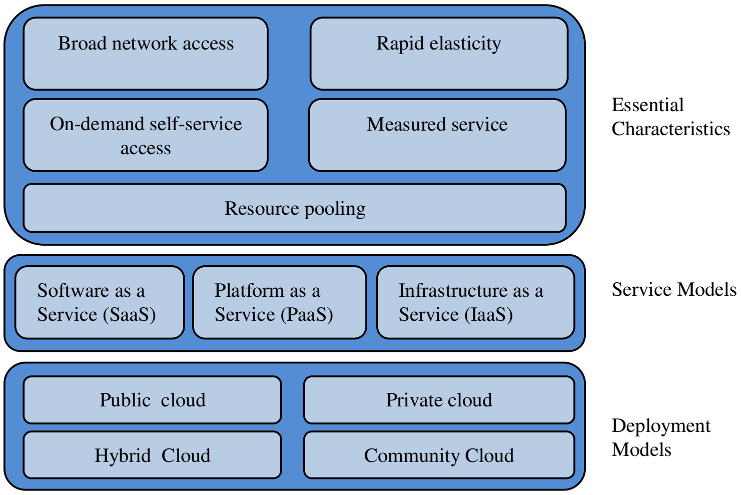

Figure 2. The NIST visual model for cloud computing
http://www.csrc.nist.gov/groups/SNS/cloud-computing/index.html

NIST also defines three service models (*Software as a Service (SaaS)*, *Platform as a Service* (PaaS), and *Infrastructure as a Service IaaS)*) which are a Service-Oriented Architecture (SOA) that describe types of services provided by the cloud. SaaS provides complete applications on CC environment. With PaaS, customers can develop, test, host, deploy and maintain their applications in the same integrated environment. With IaaS, customers use cloud infrastructure according to their demands for particular time and pay only for what they use. In addition, NIST describes a number of essential cloud computing characteristics which includes on-demand self services, broad network access, resource pooling, rapid elasticity and measured services [8,9]. With these service models and characteristics, CC implies great flexibility and availability of computing resources at different levels of abstraction at a lower cost. Cloud Service Providers (CSPs) (e.g., Google, Microsoft, Amazon) are vendors who lease to their customers cloud computing resources and software that are dynamically utilized based on customer's demand. General services in different application areas are provided to the customers online and are accessed through a web browser, while data and software programs are stored on the cloud servers located in the datacenters. This research aims at providing an infrastructure that embeds the power of CC computing resources and services into pervasive computing environment in order to overcome some of its deficiencies and to enhance its services.

4. THE PROPOSED INFRASTRUCTURE (UBICLOUD)

We propose an infrastructure for pervasive (ubiquitous) computing environment empowered by cloud services, we call this infrastructure "*UbiCloud*". We also provide a protocol that organizes communication between environment's entities. Figure 3 shows the UbiCloud environment, the infrastructure is comprised of two main components, the CC component and the Ubiquitous Computing (UbiComp) component.

- ***The CC Component:*** this component represents the cloud with all its service models (IaaS, PaaS, SaaS). It provides users in the pervasive computing environment with all services they need, for example, if the pervasive computing environment is a mall, the Restaurant Service Provider (RSP) can help user find an Italian Pizza restaurant, on the other hand, if a user is looking for the stores that have sales or bargains, the Sales Service Provider (SSP) can help her find a good deal. It is the responsibility of Cloud Service Providers (CSPs) to offer these services to the users based on the Service Level Agreements (SLA) [16]. However, these services should also satisfy the pervasive computing environment characteristics such as invisibility, adaptability and localized scalability.
- ***The UbiComp Component:*** this component represents the pervasive computing environment. It includes the following entities:
  - ***Service Directory (SD)***: which is a directory for all services offered in UbiCloud
  - ***User Agent (UA):*** this refers to any embedded device that receives service request from the user and sends it to SD. It is an interface between the mobile user and SD.
  - ***Mobile user (mobUser):*** who is any user who moves in the smart place or from one smart place to another and uses a mobile device to interact with UA.

  The protocol that organizes interactions between these entities inside the UbiCloud environment can be described as follows:

  - At the beginning, each SP in the cloud registers its service at the SD. This can be done via a *service_register* message sent from SP to SD and an *ACK* message sent from SD to SP. Hence, SD contains all services provided by the cloud for this smart place and knows which SP provides each service.

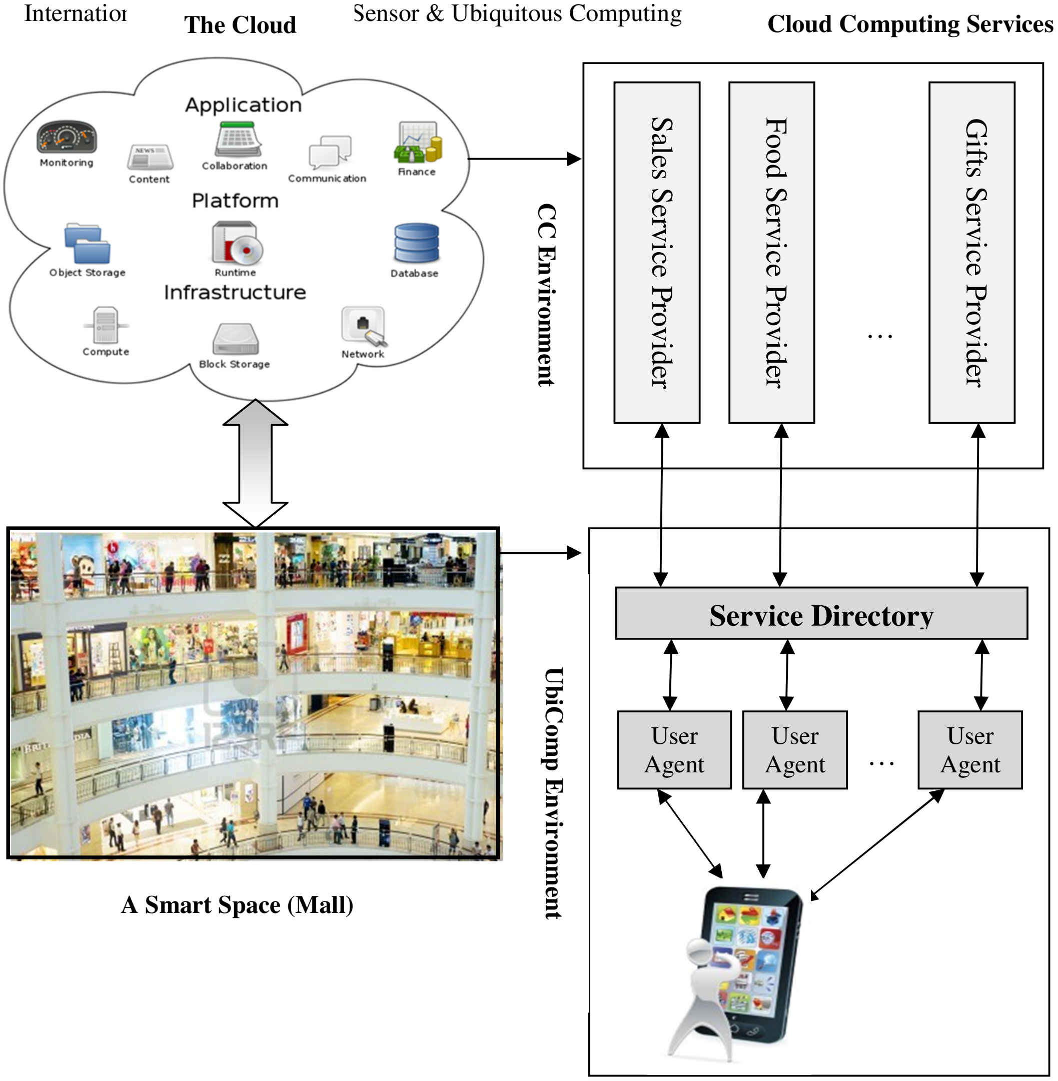

Figure 3. The infrastructure of UbiCloud Environment

- On startup, UA identifies users' preferences and intentions from the information on their mobile device.
- SD creates a service list for the smart space and broadcasts it via *service_list* message to all UAs located in this smart place. Later on, if a new SP registered a service at SD, it updates the service list and broadcasts the update list to all UAs.
- Based on users intention and preferences UA suggests some services that may be of interest to a specific user and sends it with the service list to mobUser via the message *id_service_list/suggested_sevices*. These steps are illustrated in Fig.4.
- When the user request a service from UA. The interaction between SD, UA, and mobUser is shown in Fig. 5 and is described below:
  1. The mobUser uses his/her mobile device to send a *service_request* message to UA.
  2. UA adds a user identity header to the message (*id_service_request*) and forwards it to SA.
  3. SA knows which SP provides this service on the cloud, thus, it forwards the message to the appropriate SP.
  4. When SP receives service request message from SA, it replies with *id_service_response* to UA

5. UA forwards the service response message to the mobUser.

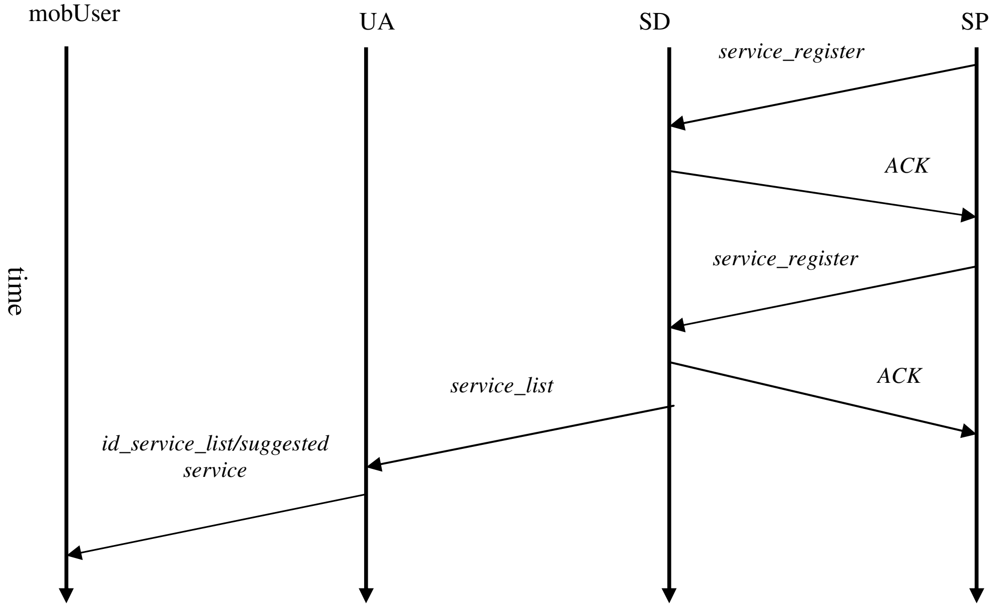

Figure 4. Initially, SPs register to SD and SD broadcasts a service list to UA which broadcasts it to mobUsers with suggested services

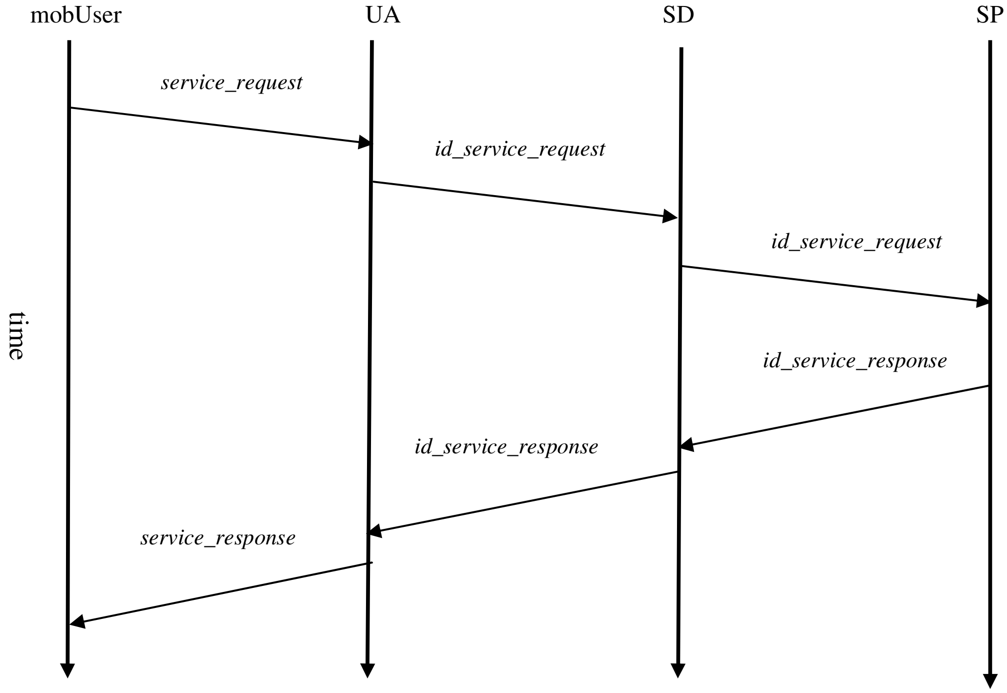

Figure 5. A mobUser request a service form UA and the UA respond with this service

The cloud services registered at SD are relevant to the smart space, i.e., when user moves from one smart place to another, a different service list is registered at SD of the new place, this helps achieve local scalability and adaptability. This concept is depicted in Fig.6. It is clear that UbiCloud infrastructure contributes to solve some inherent problems in pervasive computing. With the power of computing, storage and networking capabilities of the cloud, intensive applications that are time sensitive or need large storage space or wider bandwidth can now run in pervasive computing environments. With the high scalability of CC, more computing resources can be added, therefore; more services can be provided by CSPs to more users in the pervasive computing environment. These service are available as users move inside or across pervasive environments at anytime, since they are provided by the cloud.

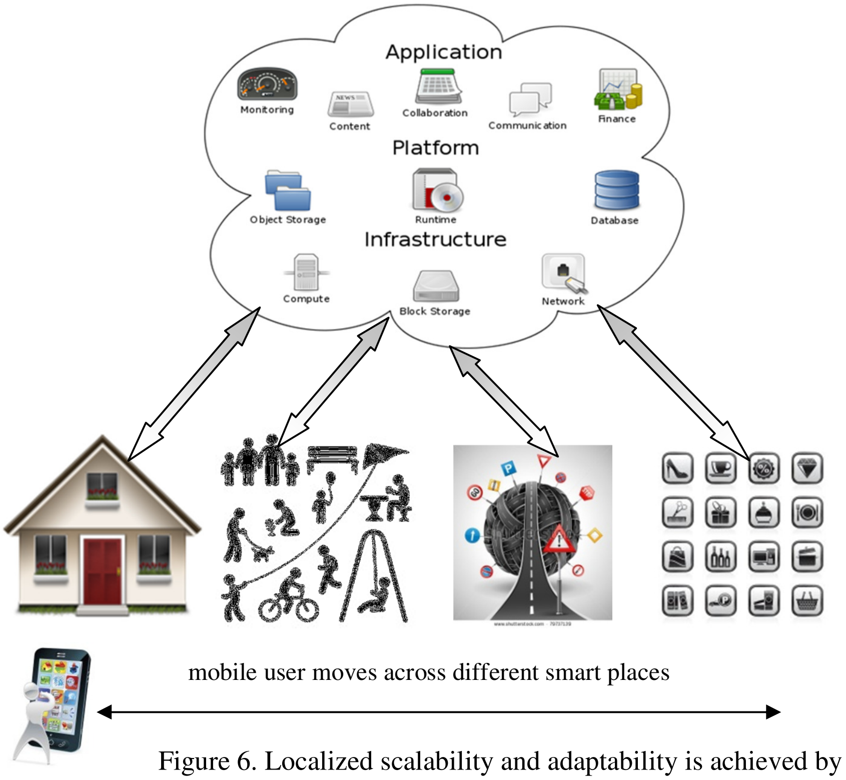

Figure 6. Localized scalability and adaptability is achieved by changing cloud services as the user moves across smart spaces

5. Conclusions and Future Work

In this paper, we examined how powerful computing capabilities, high scalability and availability of CC contributed to solve some inherent problems in pervasive computing environments. We proposed an infrastructure, called UbiCloud, that incorporates pervasive computing environments with cloud services and described a communication protocol for this environment. We showed that by embedding the power of CC services into pervasive computing environments we significantly enhance services provided by these environments. UbiCloud infrastructure is suffering from some challenges that need a lot of effort to be addressed. Security, privacy, and trust issues are of significant importance in pervasive computing. Traditional method used to provide security for individual computers and small networks such as authentication and access control are in adequate for pervasive computing environments since they lack central control and have flexible topologies. Pervasive computing systems require security architectures based on trust rather than just user authentication and access control. In the future, we plan to investigate this problem with the goal of proposing a trust model for pervasive systems. Other research issues that we would like to work on are: Service Discovery Protocols and Data Management in pervasive computing.

REFERENCES

[1] M. Satyanara Yana, (2001). “Pervasive Computing: Vision and Challenges”, IEEE Personal Communication, pp10-17.

[2] Azzzedine boukerche & Yonglin Ren, (2008). “A trust-based security system for ubiquitous and pervasive computing environment”, Computer Communications, Vol. 31, pp4343-4351, Elsevier.

[3] Shuxin Yin, Indrakshi Ray & Indrajit Ray, (2006). “A trust model for pervasive computing environments”, Int. Conference on Collaborative Computing: Networking, Applications and Worksharing.

[4] Niroshinie Fernando, Seng W. Loke, & Wenny Rahayu, (2013). “Mobile Cloud Computing: A survey”, Future Generation Computer Systems, Vol. 29, pp84-106, Elsevier.

[5] M. Weiser, (1991). “The Computers for the 21st Century”, Sci. Amer.

[6] Kevin Li. “Pervasive Computing”, University of Ottawa, Canada.

[7] NIST, http://www.nist.gov/itl/cloud/index.cfm

[8] Ahmed E. Youssef, (2012). “Exploring Cloud Computing Services and Applications”, Journal of Emerging Trends in Computing and Information Sciences, Vol. 3, No. 6.

[9] Ahmed E. Youssef & Manal Alageel, (2012). “A Framework For Secure Cloud Computing”, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No. 3.

[10] K. Hasizume, D. Rosado, E. Fernandez-Medina & E. Fernandez, (2013). An analysis of security issues for cloud computing”, Journal of Internet Services and Applications, Springer.

[11] T. Dillon, C. Wu & E. Chang, (2010). “Cloud Computing: Issues and Challenges”, 24th IEEE International Conference on Advanced Information Networking and Appications.

[12] Ahmed E. Youssef & Manal Alageel, (2011). “Security Issues in Cloud Computing”, GSTF International Journal on Computing , Vol.1, No. 3.

[13] Dimitrios Zissis & Dimitrios Lekkas, (2012). “Addressing cloud computing security issues”, Future Geberation Computer Systems, Vol. 28, pp. 583-592.

[14] S. Subashini & V. Kavitha, 20110. “ A survey on security issues in service delivery models of cloud computing”, Journal of Networks and Computer Applications, Vol. 34, pp. 1-11.

[15] P. Mell & T. Grance, “The NIST Definition of Cloud Computing” Recommendation of NIST, Special Publication 800-145, 2011. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

[16] Andrew Joint & Edwin Baker, (2011). “Knowing the past to understand the present- issues in the contracting for cloud based services”, Computer Law and Security Review, Vol. 27, pp 407-415.