



GLOBAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY: B
CLOUD AND DISTRIBUTED

Volume 14 Issue 4 Version 1.0 Year 2014

Type: Double Blind Peer Reviewed International Research Journal

Publisher: Global Journals Inc. (USA)

Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Mobile Cloud Computing

By Dr. K. V. Daya Sagar, B. Monika & K. Tejaswi

K L E F University, India

Abstract- Cloud registering can be viewed as a model that gives system access to an imparted pool of assets, such as Storage and computing power, which can be rapidly provisioned and released with minimal management effort. This paper portrays an examination action in the region of portable cloud registering. It highlights diverse open issues which are related with the portable utilization of cloud processing. By making a list of criteria for those issues, diverse arrangements are compared against each other. The solutions discussed in this paper concentrate on diverse parts of cloud processing in relationship with portable utilization. Each of the exhibited arrangements offers atleast one satisfactory approach for one of the open issues that are related with the portable utilization of cloud processing assets. By consolidating the diverse existing approaches it would be possible to create an answer that covers most of the issues currently identified.

GJCST-B Classification : C.2.4 D.4.7



Strictly as per the compliance and regulations of:



Mobile Cloud Computing

Dr. K. V. Daya Sagar^α, B. Monika^σ & K. Tejaswi^ρ

Abstract- Cloud registering can be viewed as a model that gives system access to an imparted pool of assets, such as Storage and computing power, which can be rapidly provisioned and released with minimal management effort. This paper portrays an examination action in the region of portable cloud registering. It highlights diverse open issues which are related with the portable utilization of cloud processing. By making a list of criteria for those issues, diverse arrangements are compared against each other. the solutions discussed in this paper concentrate on diverse parts of cloud processing in relationship with portable utilization. Each of the exhibited arrangements offers atleast one satisfactory approach for one of the open issues that are related with the portable utilization of cloud processing assets. By consolidating the diverse existing approaches it would be possible to create an answer that covers most of the issues currently identified.

I. INTRODUCTION

Smart phones and other mobile devices are heavily used in today's world and still get even more important since the usage of mobile internet. The development of the number of uses accessible for those gadgets in the most recent years has appeared there is a high request for versatile applications. However one common problem that all those devices share, still needs to be addressed: the limited capabilities of the gadgets in regards to accessible assets similar to processor power, accessible memory and especially energy consumption.

A technology recently emerged in the it industry offers an opportunity to tackle those issues: Cloud figuring (CC) gives its users the possibility to host and deliver services over the internet by dynamically providing computing resources[4]. Cloud registering disposes of the prerequisite for clients to plan ahead for getting diverse assets, such as capacity what's more processing force, and subsequently, is alluring to business holders. Moreover, endeavors can give assets depending on administration request specifically, assets can be dynamically added and released depending on service demand and with minimal management effort. The accessibility of cloud figuring administrations in a versatile environment, likewise called versatile distributed computing, might thus be a conceivable

arrangement for the prior specified need of resources of mobile devices. However research still needs to be done in place to unravel a few open issues similar to disclosure of cloud figuring assets, session network, also as conceivable schemas to help cloud figuring on portable gadgets.

This paper conveys an understanding into how cloud figuring systems can be utilized to help versatile gadgets and which open issues are associated with it. moreover, this paper focuses on two of these open issues, i.e. portability and asset revelation also as versatility and session integration and shows how these open issues could effectively be solved.

The main research question of this paper is:

Main research question:

Which open issues also conceivable solutions exist on using mobile cloud computing techniques to support smart phones and other resource starved devices?

To reply this inquiry, it has been partitioned into three separate subquestions, which will be answered throughout the research:

Rq1: Which cloud processing administrations can be utilized by smart phones and other resource starved devices?

Rq2: What are the open issues related with versatile cloud computing techniques to support smart phones and other resource starved devices?

Rq3: Which arrangements are tending to furthermore productively settling the asset disclosure what's more session integration open issues that are related with versatile cloud processing systems?

Most of these research questions will be answered using a writing study. The third research question (Rq3) utilizes a combo of writing study, qualitative examination and determination of new characteristics to be coordinated into versatility resource discovery and session connectivity methods.

The remainder of the paper is organized as follows. section II quickly describes the available cloud computing. Section III talks about a few open issues that are connected with versatile cloud registering and their answers. The examination and comparison of the different solutions is presented in section IV. Section V concludes and provides recommendations for future exercises.

Author α: Associate Professor, Department Electronics and Computer Engineering, K L E F University, Vaddeswaram, Tadepalli (M.D), Guntur (Dist), India.

Author σ ρ: B.Tech, Department Electronics and Computer Engineering, K L E F University, Vaddeswaram, Tadepalli (M.D), Guntur (Dist), India. e-mail: tejacool987@gmail.com

II. CLOUD COMPUTING SERVICES

The national institute of standards and technology (nist) defines cloud computing as "a model for enabling convenient, ondemand system access to an imparted pool of configurable processing assets (...) that can be quickly provisioned and discharged with insignificant administration exertion or administration supplier connection". Agreeing to NIST, key characteristics of cloud computing are:

- Ontdemand selftservice
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service

On the demand selftservice implies that assets can be requested and released without human interaction at the service provider s side. furthermore, those resources are accessible via institutionalized system conventions and are pooled at the administration provider s site. the resources are dynamically assigned to the diverse clients that need to make utilization of them, with the side effect, that the user does not exactly know where the resources he/she at present utilizes are located. the dynamic task of resources can be done manually or automatically, whereas the last offers the possibility to quickly react on resource demands and scale in or out accordingly. the usage of resources can be checked to control and upgrade it individually. Other than those attributes, cloud figuring administrations can be classified into three different service models:

- Infrastructure as a service (iaas)
- Platform as a service (paas)
- Software as a service (saas)

The models will be examined in point of interest in the accompanying areas.

a) *Infrastructure as a service (laas)*

The first service model, which is called infrastructure as a service , is based on the provisioning of figuring assets which are more fittings situated. Agreeing to NIST the provisioning of "transforming, stockpiling, systems, and other principal figuring assets where the consumer is capable to convey and run subjective programming, which can incorporate working frameworks and applications", from fall under this classification. With infrastructure as a service, the user is able to run furthermore deal with own working systems including applications by using virtualization technologies. furthermore he can make use of capacity frameworks and/or system devices like e.g. firewalls.

Samples for this type of service model are amazon ec2 for reckoning force and Amazon S3 for

capacity provisioning. With a view to mobile environments, this service model does not seem to be appropriate for mobile usage of cloud services, as it is very concentrated on the procurement of equipment based administrations with a low level of deliberation. It may just be intriguing in the case of capacity provisioning for portable gadgets.

b) *Platform as a service (paas)*

The second administration model, which is called "stage" as a service , gives users the opportunity to run applications on the framework offered by the administration supplier. However, it obliges that the applications are made with programming dialects or tools that are backed by the administration supplier. Illustrations for this sort of administration model are Google App Engine, force.com and microsoft windows azure. Looking at portable usage of cloud registering services, this service model seems to be of interest, because it gives users the possibility to outsource applications or parts of them to the cloud. As a result, clients can make utilization of the profits a cloud processing system can offer them, including scalable and fast computation resources, which in the end could save time and energy

c) *Software as a service (saas)*

The third service model, named software as a service focuses on the provisioning of uses. Cases for this type of administration model are Google Docs, microsoft Office web Apps also Apple iwork.com. With respect to portable use, this administration model may additionally be of investment, in spite of the fact that it completely relies on s upon a working system association between portable gadgets and the cloud system. however, the benefits of cloud frameworks, which may lead to funds in time and vitality consumption, also apply here

III. OPEN ISSUES AND SOLUTIONS

The mobile usage of cloud computing services is still in the early stages of development and several open issues need to be tended to. With the mobility of users and their devices, several problems arise that need to be taken into account, when making use of cloud computing services on mobile devices.

a) *Mobility and resource discovery*

The first being, that cloud computing resources are widely spread around the globe and offer a lot of different services to their clients. Mobile gadgets that need to make utilization of those assets ought to be capable to consequently find cloud processing assets that ideally are adjacent their current area.

b) *Mobility and cloud session connectivity*

Another problem to face is the fact that mobile devices are not generally s joined to a system the grounds that of dead spots or other impacts,

coming about in inaccessible cloud processing resources. related to that is the fact that connectivity with remote cloud computing resources can suddenly disappear. Furthermore, network addresses of mobile devices can change over time, due to diverse local address assignments, coming about in a need for topologyagnostic distinguishing proof of associations.

c) *Programtechnical characteristics*

Offloading of calculation to remote assets is a procedure that obliges extra exertion and abilities of application engineers, as applications conceivably need to be adjusted accordingly. to make full use of remote resources, it may for instance be fundamental that parts outlined for remote execution need to be diverse from their neighborhood supplements, as they need to make utilization of parallelization, which also demands additional skills from developers overhead due to use of cloud .

Along with the remote execution of application parts comes the problem that those parts need to be transferred to the cloud resource first, before an execution can take place. the overhead delivered by this transfer also needs to be taken into account, when dealing with computation offloading and possibly related time and energy savings.

d) *Unwavering quality*

The ability of the cloud computing system to perform and keep up the procurement of its assets under sudden disappointments, of e.g., stockpiling, system network and figuring power, for a predefined sum of time. This capacity can be upheld by e.g.,

- 1) supporting replication of objects and administrations,
- 2) utilizing repetitive correspondence (more than one correspondence ways utilized for the scattering of the same data),
- 3) utilizing repetitive handling (more than one processing entities used to process the same action).

e) *Versatility*

The capacity of the cloud figuring system to maintain its great framework execution while supporting

- 1) an expanding amount of mobile users,
- 2) expanding the amount of resources and services to satisfy rapid increases in service demand. this ability can be satisfied by e.g.,
 - support for massive sharing of content,
 - flexible, faulttolerant and distributed data bases,
 - ast and consistent content replication support.

f) *High availability*

The capacity of the cloud processing system to give and help a substantial sum of diverse

assets that are effectively available and that are working in ideal execution conditions for a predefined agreed amount of time.

g) *Security and privacy*

The ability of the cloud computing system to protect itself furthermore its given assets from security and protection assaults. Diverse security and protection viewpoints need to be considered at the point when running outside code on remote assets that perhaps likewise utilized by a few clients at the same time. The fundamental security arrangements are e.g., related to

- 1) data integrity, where the unapproved change of data approaching and cordial the cloud ought to be located,
- 2) classifiedness to secure the data access and transfer. the main privacy solutions should ensure that the identity of the cloud processing customers should not be revealed to unauthorized entities.

Due to paper page limitations, only the open issues subset that is the most relevant for mobile cloud computing is chosen for further investigation:

- 1) Mobility and resource discovery
- 2) Mobility and cloud session connectivity.

a. *Criteria*

The accompanying criteria are related with the open issues mentioned above.

1) *Mobility and resource discovery*

The criteria related with the open issue "versatility and resource discovery" are defined as follows:

Naming and addressing of resources (ma): The resources are using a naming/addressing method and/or structure. Three grades are utilized for rating this measure:

- i. Great: Supportedg
- ii. Moderate: Supported yet problematicg
- iii. fair: not supported.

Dynamic discovery of cloud resources (rdd): The resources can be discovered using a dynamic method. Three grades are utilized for rating this measure:

- i. Good: Dynamic discoveryg
- ii. moderate: manual discoveryg
- iii. fair: Static discovery.

Latency of the resource discovery process (rdl): The sum of time devoured for the disclosure of available cloud computing resources. two grades are used for rating this measure: "

- i. Good: Time accomplished by an automated discovery processg

- ii. fair: time experienced by a manual discovery process.

2) Mobility and cloud session connectivity

The criteria related with the open issue "versatility and cloud session connectivity" are defined as follows:

Handover support between resources (hsr): The ability to migrate from one cloud computing resource to another. Two grades are utilized for rating this model:

- i. good: supported
- ii. fair: not supported

Latency of handover between resources (hhr): The amount of time used for the handover process from one asset to an alternate. Three grades are utilized for rating this model:

- i. Good: Seamless handover
- ii. Moderate: Short interruption
- iii. fair: long interruption.

Data loss probability during handover (hdlp): The likelihood that information misfortune s will happen amid the handover from one resource to an alternate. Two grades are utilized for rating this criterion:

- i. good: no data loss at all
- ii. fair: High chance for data loss.

Roaming support for mobile devices (rs) : The capacity to help meandering of versatile customers through distinctive remote s system advances. The arrangement is considered to be efficient when:

- a) The identification of connections is topologyagnostic
- b) The solution supports a seamless handover procedure

Three grades are used for rating this criterion:

- i. good: (a) also (b) are supported
- ii. Moderate: Either (an) or (b) is supported
- iii. fair: neither of the two is supported.

b. Evaluation of solutions

The assessment of arrangements and the degree in which the distinctive arrangements match the criteria is based on the documentation and assets accessible for the specific solution. table I gives an outline over the images used in the scoring tables and their meaning.

i. Cuckoo

Cuckoo is a processing offloading system for cell phones which, at the minute of composing this archive, has been executed for utilization with the Android working system only. Cuckoo has been designed with the fact in mind that cloud processing assets are not generally accessible when utilized by

versatile gadgets. Therefore its modifying model helps both local and remote execution of application methods to keep applications working when cloud resources are not available. it supports different implementations of application methods for nearby and remote locales to secure the use of offers that may only be available in cloud resources (e.g. parallelization).

Cuckoo has been actualized as a standard server/customer model. the server can run on any resource, which has a Java virtual machine installed. services (parts of an application) that are accessible on the customer can be transferred to the server and executed remotely a while later. When an administration has been at first transferred to the server, it can be utilized a while later without the need to upload it again. The discovery of resources is done manually in the current prototype. as soon as a server is running, a qr code (short for "Fast Response", a twotdimensional standardized tag) is shown, which can be scanned in by a client with its integrated camera. The asset (or more precisely its address) is then stored in the "Asset Manager", a part of the Cuckoo schema, which is capable for connection for the responsibility.

The asset can then be used to offload and execute parts of an application.

The procedure s of choosing whether a part of an application should be executed local or remotely, is done by making use of heuristics, context information and history. At the moment of composing, these heuristics noticeably are extremely straightforward: remote execution is preferred over local execution, if remote resources are available.

Communication between server and clients is realized with the Ibis correspondence middleware, which modified works and supports different networks like wifi, cellular and bluetooth.

The Ibis correspondence middleware is a versatile, high performance, java based library. Due to the attributes of versatile situations, connections to resources can be lost over time. cuckoo handles these detaches by exchanging to diverse assets and proceeding with execution on another accessible asset. As a last fallback, execution of use parts can simply be carried out locally on the mobile device. The current asset revelation procedure s of Cuckoo can be considered wasteful, on the grounds that of the actuality that for every new asset, a manual examining of the resource s QR code is needed, see table ii. this takes up a large amount of time and is not a very scalable process. Cuckoo supports migration from one asset to an alternate and has a nearby fallback that empowers applications to stay working, when no cloud assets are accessible. The methodology s of relocation itself be that as it may is definitely not portrayed, so there is no confirmation

of right living up to expectations or information about the latency of this process. due to the use of the Ibis communication middleware, it is possible to establish correspondence between portable gadgets and cloud administrations over different networks.

ii. VOLARE

Volare is a middle ware based solution, which gives context aware versatile cloud administration disclosure for portable frameworks. It does this by observing assets and connection of gadgets and alertly adjusts cloud administration demands accordingly. Operation takes place at two levels: service discovery time and runtime. At administration disclosure time, VOLARE catches administration demands from applications executed on a versatile gadget. Agreeing to the current connection of the gadget, which might incorporate equipment assets (e.g. battery utilization, CPU, memory use), natural variables (e.g. system data transfer capacity) and client inclinations (e.g. low cost tying, low power operation) it then starts to process the request. At runtime, volare continuously monitors existing cloud ties and the connection of a gadget. If the setting of the gadget or the gave administration level of the cloud administration changes, VOLARE responds on this via seeking for an administration that matches the new requirements and initiates a rebinding. The building design of VOLARE comprises of a few independent modules. if an application requests a service, the solicitation is blocked by the administration demand module and sent to the adaption module. At the same time, the context of the device is continuously monitored by the context monitoring module and information about the current context is sent to the adaption module. The adaption module handles the service requests according to the current context of the device. If changes in the context occur, a reevaluation of presently dynamic benefits is activated, conceivably coming about in a rebinding agreeing to the new connection and Qos (Quality of Service) level. The satisfaction of Qos levels is checked by the Qos checking module. If deviations are stated, the service request module gets alerted, resulting in the initiation of a new discovery cycle. the service discovery and rebinding is done by the administration tying module. It advances the adjusted administration appeal to a representative, which then picks the best matching service provider. Concurring to , the structural engineering empowers the utilization of VOLARE without modifications to the application itself. The definition of qos levels takes place by providing an "adaption strategy record", which is composed in an own two level approach specification language, together with the application. Volare uses brokers to access services in the cloud. the handle acknowledges

an administration demand with a relating administration level and tries to tie to the best matching administration supplier. VOLARE helps handover from one asset to an alternate. Agreeing to, the "rebinding keeps going s a normal of 0.963 seconds", from , which is considered a very low latency, see

Table ii. however, it should be mentioned that due to the fact that VOLARE has not concentrated on remote execution of applications, there was no need to move (application) information between assets. Methods for topology agnostic ID of associations did not appear to be connected, as communication took place through standard tcp/Ip protocols (the prototype used RTP/Rtmp for video streaming).

iii. Mobile computation outsourcing framework

Chonglei Mei[2] proposes a portable processing outsourcing skeleton for the Android stage, which is actualized in Java and agreeing to it can be effortlessly deployed on any backend platform. The framework consists of three main components, which are located at the cloud side: a proxy, a code repository and a server. The proxy server acts as a gateway between mobile devices and the cloud. it has access to a code repository, which contains popular code components in the cloud "When" a mobile device "wants" to offload computation to the cloud it contacts the proxy server with the name of a particular Java class component it wants to execute. According to, three cases need to be differentiated:

- 1) the code might already been running on one of the servers, so the mobile device can make use of it immediately
- 2) the code might also be available in the code repository, but has not been deployed on any of the servers, thus the proxy server can push the code to one of the servers and begin itg
- 3) if the code is not one or the other running nor accessible in the vault, the versatile device needs to upload it accordingly. communication between versatile gadgets and servers, which execute the code, takes place directly.

For asset discovery, proposes to place the substitutes outside the cloud, so that they can run a multicast DNS (mdns) to advertise their services to versatile devices in their system. This is carried out by TV data about the offered service type and name to neighbors. the proxies could be placed at e.g. wifi access points. if the proxies are placed at the cloud side, proposes to partner a well known URL with the proxy, so that devices can connect to it.

At the versatile device s side, a customer is dependable for administration of reckoning offloading. The customer screens accessible assets of the versatile gadget and makes a execution

history of offloaded application segments. Based on both information it then chooses whether to offload computation to the cloud or not. The mobile computation offloading framework uses dns furthermore particularly mdns to address cloud processing assets. Due to the large numbers of cloud computing resources this is considered wasteful, as the current DNS structure might get to be over-burden. Therefore, this answer is appraised as moderate, see table II. discovery of resources is achieved by putting substitute servers nearby to system access focuses, which publicize cloud administrations through mdns. This is considered to be great, as it offers an element path to find assets and gives clients the likelihood to be redirected to assets close their current area. Mechanisms for handover or wandering support could not be identified.

IV. TALK

The arrangements introduced in this paper concentrate on diverse parts of cloud computing in association with portable use. Table II gives an outline over the distinctive arrangements and their rating. The Cuckoo structure introduces an answer to offload application parts to cloud registering assets. It handles conceivable integration problems with the help of local fallback methods and offers components for handover help between resources. VOLARE with its middleware methodology offers context mindful versatile cloud administration disclosure for portable frameworks. Services are chosen based on the current context of the device what s more at present gave administration levels are ceaselessly observed and thought about against the prerequisites. If deviations are asserted, volare reacts on it by searching for a more appropriate service and initiates a rebinding. The Mobile Computation Offloading Framework proposed by mei et al. (2011) offers an interesting approach on how to find close-by cloud figuring assets with the help of multicast dns. Current cloud processing bases tend to be outlined for utilization with wired or settled gadgets fundamentally. Nonetheless, due to the quick improvement of remote s broadband web associations for mobile devices, this outline has to be changed and mobile devices need to be considered as parts of those cloud processing bases too in the close future. Several open issues that are connected with the coordination of those versatile gadgets into cloud figuring foundations, are still not yet agreeably being tended to by current cloud processing arrangements. One sample is the to pologytagnostic identification of mobile devices, which tend to change their location (and therefore their current ip address) more often.

V. PROPOSED SOLUTION

None of the talked about arrangements can fulfill all the criteria related to mobility and resource discovery as well as mobility what s more cloud session connectivity in a fitting way. therefore, in this segment we will present distinctive approaches that can enhance the solutions discussed above.

a) *Naming and a dressing of resources (rna)/ Roaming support for mobile devices (rs)*

A conceivable arrangement that can fulfill these criteria is described in the internetdraft "locator/Id separation protocol (Stutter)". Stutter is a network based convention that empowers tending to of gadgets autonomous from their topological area by dividing IP addresses into two new numbering spaces: Endpoint Identifiers (Eids) and Routing Locators (Rlocs)[1]. It utilizes Eids to recognize gadgets autonomously from the system topology and Rlocs, which are topologically allotted to system connection focuses, for routing and forwarding of packets through the network. Stutter has been planned with straightforward and incremental deployment in mind, meaning that no changes to current host protocol stacks or core internet infrastructure are needed. When a Lisp enabled router receives a packet with a nont routable eid assigned to it (as destination address), it maps the EID to a routable RLOC and typifies the bundle for further sending. The bundle can then be sent to a Stutter empowered switch at the goal by nontlisp routers. The LISP empowered switch at the end of the line decapsulates the packet and forwards it to the recipient specified by the eid. With this methodology it is conceivable to create stable correspondence between gadgets free of their current geological and topological area and to unravel the connectivity problems mentioned before.

b) *Dynamic discovery of cloud resources (rdd)*

The Universal Description, Discovery, and Integration (UDDI) protocol is a standard created by the organization for the Advancement of Structured Information Standards (Oasis). uddi defines a "standard method for publishing and finding the network based programming parts of a service toriented architecture[3]". It acts as a registry for web services on public or private networks by offering information about accessible administrations with the assistance of guidelines similar to e.g. XML. Like web administrations offer some sort of administration to their customers, cloud figuring administrations do so in the same way with assets. UDDI, or an altered rendition of UDDI more fitted to cloud registering may thusly be a fascinating approach to build an institutionalized way for the disclosure of cloud computing resources.

VI. CONCLUSION AND FUTURE WORK

In segment II of this paper, a presentation into cloud registering and its diverse administration models has been given in order to answer the first research question (Rq1), which was intended to show which cloud computing services can be used by smart phones and other resource starved devices. The three diverse administration levels have been dissected with respect to mobile usage, where IaaS with its hardware oriented approach only seems to be appropriate for storage provisioning. The two other service levels, namely PaaS and SaaS seem to be of more interest, as they offer the possibility to run whole applications or parts of them in the cloud. In section III, several open issues that need to be taken into account, when making utilization of cloud processing benefits on versatile gadgets, have been portrayed in place to reply the second research question (Rq2). Furthermore, different criteria have been established in order to assess arrangements for those issues. This comes about in a portrayal of right now created arrangements that (somewhat) address the open issues specified before and gave an answer to the last research question (Rq3). Each of the exhibited arrangements offers at minimum one palatable approach for one of the open issues that are connected with the versatile utilization of cloud processing resources. By combining the different approaches and merging them into a common solution, it might be possible to provide a new arrangement that covers most of the open issues as of now identified. Such a solution might have the possibility to finally make cloud figuring usable on mobile gadgets, resulting in new and fascinating utilization situations and offering execution speedups and energy savings to mobile users.

REFERENCES RÉFÉRENCES REFERENCIAS

1. Farinacci, D., V. Fuller, D. Meyer, and D. Lewis, Locator/ID Separation Protocol (LISP). Internet Draft, July 9, 2011. Available online at: <http://tools.ietf.org/id/draft-ietf-lisp-15.txt>.
2. Mei, C., J. Shimek, C. Wang, A. Chandra, and J. Weissman, Dynamic Outsourcing Mobile Computation to the Cloud. 2011, Department of Computer Science and Engineering, University of Minnesota: Twin Cities.
3. Organization for the Advancement of Structured Information Standards (2004). "Introduction to UDDI: Important Features and Functional Concepts." Retrieved July 19th, 2011, from <http://uddi.org/pubs/uddi-tech-wp.pdf>.
4. Zhang, Q., L. Cheng, and R. Boutaba, Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 2010. 1(1): p. 7-18.

This page is intentionally left blank