Journal of Theoretical and Applied Information Technology
15th August 2016. Vol.90. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SERVICE DISCOVERY IN AD-HOC MOBILE CLOUD: CONTEMPORARY APPROACHES AND FUTURE DIRECTION

[1]DOMINIC AFURO EGBE, [2]M.B. MUTANGA, [3]M.O. ADIGUN
[1,3]Department of Computer Science, University of Zululand, South Africa
[2]Department of Information and Communication Technology, Mangosuthu University of Technology, Durban, South Africa
E-mail: [1]domafuro@gmail.com , [2]mutangamb@mut.ac.za, [3]profmatthewo@gmail.com

ABSTRACT

Service discovery is fundamental to Service Oriented Computing (SOC). Ad-hoc Mobile Cloud (AMC) takes advantage of SOC and Service Oriented Architecture principles in providing services to clients. This emergent paradigm promises to alleviate mobile devices from challenges associated with consuming remotely offered services. However, like other service offering platforms, AMC faces additional and unique service discovery challenges due to their inherent nature such as mobility, resource-constraints and dynamic context. Although conventional web services technologies and approaches exist, they do not fulfill the core requirements of mobile devices, and hence are not suitable for use in the AMC environment. To facilitate the realization of the prospects of AMC, it becomes highly imperative to study conventional and emerging web services approaches and their impact on performance in AMC domain. In this paper, we study existing web services enabling technologies and approaches that impact on service discovery in AMC. We focus on suitability for AMC based on resource requirements and other factors associated with mobile environments. Our findings reveal that conventional web services approaches cannot be directly applied in AMC environment. This is mainly due to their inability to meet the requirements of resource-constrained devices. On the other hand, other approaches that consider resource limitation and other challenges associated with resource-constrained environments need to expand their scope to include the application of context information.

Keywords: *Ad-hoc Mobile Cloud, Service Discovery, Architecture, Resource-intensive, Dynamic Context*

1. INTRODUCTION

A distinctive phenomenon of the past decade is the expansion of e-markets and the rapidly growing mobile consumer base [1]. This trend has prompted the emergence of the mobile web services paradigm, that is, web services that are offered from mobile devices and accessed and invoked by peer nodes. Interestingly, there is now an expanding interest in such web services derived from the surging number of mobile devices and the unprecedented advancement in the capacities of such devices [2], [3]. The envisioned prospect of mobile web services has, in recent years, scoped a new research direction aimed at advancing mobile web service provisioning as well as developing appropriate platforms to support its operation [4], [5], [6].

This emergent research focus has boosted Mobile Cloud Markets and has created the influx of e-service consumers seeking cost effective services [7]. Nevertheless, mobile consumers face the unique challenge of resource-poverty, intermittent internet disconnection, high latency, and energy consumption. These constraints exert a profound impact on the efficiency and suitability of service discovery mechanisms designed for pervasive environments that employ resource constrained devices [8], [9].

Ad-hoc Mobile Cloud (AMC) computing has recently evolved to address the aforementioned challenges. AMC provides the framework that exploits the idea of mobile web services in terms of developing appropriate hosting and discovery mechanisms. This Cloud paradigm presents a platform for inexpensive and collaborative resource provisioning. AMC is formed by a set of mobile

Journal of Theoretical and Applied Information Technology
15th August 2016. Vol.90. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

devices (nodes), each having the capability to act as service provider or client, or both.

Another feature of this computing paradigm is that participating nodes communicate over wireless mediums and operate without the benefit of any infrastructure, thereby creating an opportunistic resource sharing platform [3]. Such characteristics make AMC an ideal solution for scenarios with weak or no internet connection. Besides complementing the Cloud, AMC can minimize monetary costs with regard to Internet subscription charges as well as avert other huge capital investments associated with network and other infrastructure such as base stations, routers and switches, access points, computers and related services.

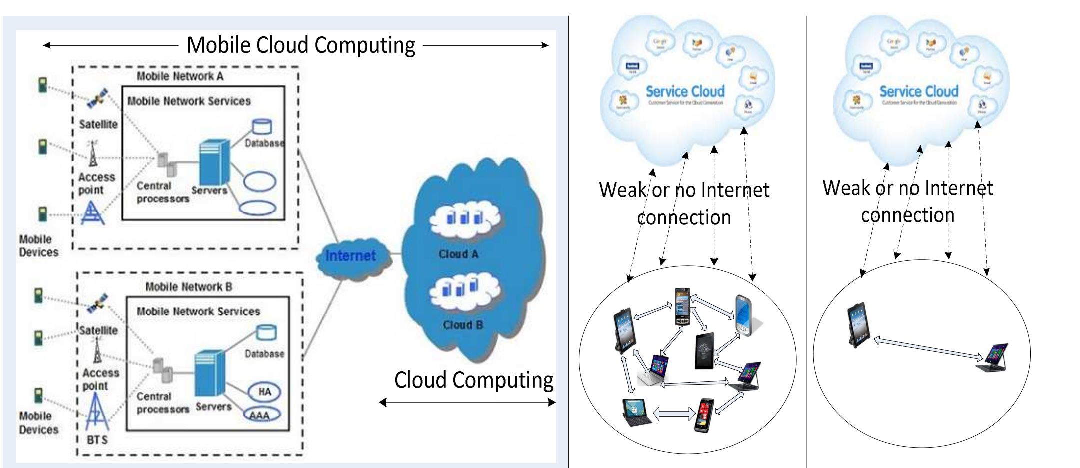
Nonetheless, the realization of the overall goal of AMC, to a large extent, depends on how best the approaches adopted in the web services process answers the core challenges of resource-scarce environment.

Existing standards for web services are either tailored for wired networks for accessing resource-rich platforms such as the Cloud and Mobile Cloud, or are enhanced with techniques that can potentially violate the resource requirement of mobile devices [10], [11]. In attempt to advance the prospect of AMC, this paper takes an in-depth look on the main web services approaches with a view to weigh their impact and suitability for deployment in AMC. We also study the current state-of-the-art in Ad-hoc Mobile Cloud service discovery, the techniques utilized to address dynamic context and resource scarcity, and point out open issues as well as possible future direction.

The remainder of this paper is structured thus: Section 2 presents the formation of AMC, whereas service discovery challenges in AMC are discussed in Section 3. Section 4 analyzes web services standards in the AMC perspective, and various web service architectures are described, and their strengths and weaknesses outlined in Section 5. The current state-of-the-art in AMC service discovery is then presented in Section 6, followed by a brief discussion in Section 7, which concludes this paper. Here we highlight our findings and discussed possible future directions. Throughout this paper, we refer to mobile web service as web services or services and mobile device as mobile nodes.

2. THE AD-HOC MOBILE CLOUD (AMC)

The Ad-hoc Mobile Cloud is a complementary approach to web service provisioning, being considered in the Mobile Cloud domain which consists, instead, in enabling direct service provisioning between mobile nodes. On this web service platform, mobile nodes exploit self-organizing networks to support direct communication between each other in order to expose their computing resources and services to others within the network [3] [12]. As illustrated in Figure 1, the Mobile Cloud enables mobile nodes to access conventional Cloud services. However, there are no guarantees for continuous and reliable internet connectivity, especially in rural areas. Even in instances where there is strong and stable internet connectivity, research shows that uplink connections lead to higher energy consumption [13], which does not favour battery-dependent devices.



"A": MCC architecture [14] "B": AMC formation "C": Ad-hoc node interaction

Figure 1: Formation of Ad-hoc Mobile Cloud from Conventional Mobile Cloud

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The AMC idea seeks to address these challenges based on the assumption that each mobile node can independently access remote Cloud services. In the event of loss of internet connection, however, the affected node can fall back to the local Cloud (AMC) to search for web services. On the other hand, a client now has an option to opt for the remote Cloud only when a desired service cannot be found within the Ad-hoc Cloud.

Figure 1A shows the Mobile Cloud Architecture, which integrates mobile devices into the conventional Cloud setup. The middle and right hand figures (1B and 1C) illustrate the formation of Ad-hoc Mobile Cloud by peer nodes. The Figure 1C further illustrates how a client node connects to a nearby providing peer to discover service.

Although in general large-scale web service hosting on mobile devices is not feasible, the AMC setting is envisioned to offer services that are tailored to the needs of small computing communities such as Small, Medium and Micro enterprises (SMMEs).

3. SERVICE DISCOVERY CHALLENGES IN AMC

Basically, AMC takes advantage of the advancements in mobile device's capabilities coupled with the exploits in wireless communication technologies. Nevertheless, such advances do not eliminate the inherent limitations of these portable devices, which invariably impact on service discovery in domains that employ such devices. Among other factors that influence service discovery in AMC, the chief challenges include:

3.1 Resource Limitation

AMC is a setup of mobile nodes and the issue of resource limitation in mobile nodes is an inherent challenge. As described in [14], mobile devices are generally said to suffer from resource poverty. That is, they have limited processing power, memory size, disk capacity, and above all, are battery dependent. These limitations become a major challenge in AMC because each node is expected to act both as service provider and client. So, resource utilization must be minimized to keep the entire system running.

Therefore, to realize efficient and affective service discovery, mechanisms intended for AMC must take these limitations into cognizance by adopting techniques that are efficient in terms of their demand for computational resources [15], [16]. Such consideration is desirable because service discovery operations are said to be resource demanding, which can infringe on the resource capacity of mobile devices [10], [11]. For example, lightweight technologies are required to avoid situations were a client or providing nodes runs out of critical resources due to computational burden [10].

3.2 Mobility

There is active research in mobile or pervasive computing because it is still an evolving field of study, whose body of knowledge is broad in context [17]. Some of the areas covered by mobile computing include: mobile networking (mobile IP, ad hoc protocols etc.), mobile information access (disconnected operation, bandwidth intelligence etc.), adaptive applications support (resource management), system-level energy saving (energy-awareness), and location sensitivity (location-awareness) [17], [14].

However, these areas emanate from the central feature of mobile computing – Mobility. This feature introduces the non-trivial challenge of designing intelligent mobility management techniques that can ensure that provider's mobility does not disrupt seamless operations. That is, maintaining uninterrupted communication between providers and clients of web services amidst intermittent and heterogeneous wireless environment [17], [19].

The above challenge potentially impacts on service provisioning and discovery, because when inter-node communication is disrupted, services cannot be accessed. To achieve effective service discovery in AMC therefore, it is expedient to adopt web services approaches that support mobility principles.

3.3 Dynamic Context

Generally, mobile environments are associated with unpredictable changes. These can be changes in local context, which may include device resources (battery, memory), environmental variables (bandwidth, Internet access etc.), or user preferences [20]. With respect to service discovery, context is described in [21] as any implicit information that can affect the usefulness of the retrieved service. That is, a change in context may in one way or the other affect service discovery in AMC.

Dynamic context requires systems with context-awareness, that is, with the ability to use contextual information to change and automatically adapt to current context. With such ability, services

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

discovered are tailored according to the context of clients. Nevertheless, utilizing context with advanced techniques is AMC is challenging because of resource scarceness, which places restriction on permissible techniques [21].

3.4 Operational Architecture

The AMC utilizes the infrastructure-less concept of Mobile Ad-hoc Networks (MANETs). This concept implies the absence of any central point of control.

In the last decade, researchers have leveraged on dissimilar architectures to implement web service provisioning in AMC. However, these different efforts focus on addressing varied challenges of interest [5]. Such diffusion can prolong standardization of AMC because of scattered development approaches without any reference architecture [22].

Most importantly, different architectures utilize varying technologies that have intrinsic features that may not conform to the resource requirement of AMC or support some concepts of mobile computing.

Hence, AMC is still faced with the challenge of the absence of a generic reference architecture that guarantees optimal service discovery performance.

3.5 Security, Trust and Privacy

The issues of security, trust, and privacy are major challenges in resource sharing domains. Such challenges are much more threatening in wireless environments due to their vulnerability to attacks.

Although service discovery may not be directly affected by some of these mentioned issues, there are still some dimension of malicious attacks such as wireless denial of service (DoS) [23] that can disrupt normal function of AMC.

Some researchers like [9] have proposed a security framework for personal web services. However, there is need to develop a unique security framework for AMC considering that the service provider is a personal device with confidential information.

4. WEB SERVICES STANDARDS: AN AMC PERSPECTIVE

The aforementioned challenges are major open research issues facing service discovery in AMC. These challenges, in a way, affect service discovery. Therefore, an effective and efficient AMC service discovery mechanism should be proactive to context change and adopt techniques in the web services process that take those challenges into account. In this work we define web service process to consist of service description and service implementation processes as depicted in Figure 2. This section discusses the implication of conventional web services techniques on AMC.

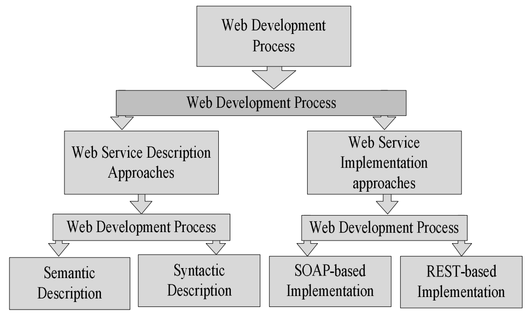


Figure 2: The Web Service Process

4.1 Web Service Description Approaches

A web service is a loosely coupled, independent, modular software component that exposes particular business functionality on the Internet for other applications to utilize via established web protocols for example, HTTP and XML [24]. Web Services is therefore a set of standards as well as programming methods aimed at sharing data between different software applications. Such programming techniques basically implement the Service Oriented Architecture (SOA) paradigm. Meaning, with SOA principles, software can be delivered over the network in the form of services.

The process of translating a software module to consumable network business functionality is referred to as service description.

Majorly, there are two approaches to web service description namely, syntactic and semantic. However, the choice of a suitable web service description approach should be informed by the requirements of the environment.

The web service description language (WSDL) is the de facto standard for describing web services. Technically, a services description covers three aspects of a web service: the information model, functional capabilities, non-functional parameters, and the technical model.

a. Information Model: In the information model providers define the data model, which is made of input/output messages as well as other data relevant to the service operation. In order to accommodate the heterogeneity of middleware platforms, programming languages, consumers and

Journal of Theoretical and Applied Information Technology
15th August 2016. Vol.90. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

other technologies, web services are designed to support interoperability [25]. This design goal is achieved using the information model, which facilitates intercommunication.

b. Functional capabilities: The operations that are offered by services and how potential consumers can interact with services are determined by functional capabilities specification. The reason for functional descriptions is to provide a black box description of what a web service does, disregarding other detailed technical information. Primarily, in traditional web services discovery mechanisms, web services are differentiated based on their functional capabilities.

c. Non-functional parameters: This specifies the running and environmental parameters such as reliability, availability and quality of service (QoS) etc. With context playing a vital role in pervasive environments, non-functional parameters specification has gained increased attention in service discovery research. It has generally been argued that enriching service descriptions with sufficient non-functional parameters will enhance support for automated service discovery, boost the discovery of relevant services, and ease web service customization [26], [27].

d. Technical model: Technical specifications are primarily concerned with details about implementation such message structures, transport protocols, access information, and service location. In the traditional web service discovery framework, technical specifications are contained in the "green pages" of the Universal Description, Discovery, and Integration (UDDI) service registry. These processes of service description all together contribute to the effective discovery of web services because for services to be discovered they have to be well described and published [28].

The realization of the entire service description process is based on the two major approaches:

4.1.1 Syntactic web service description

Syntactic or on-semantic web services are described using the Web Service Description Language (WSDL). The WSDL is an XML format for describing services that are offered over networks as a set of endpoints operating on messages [29]. With WSDL, web service providers are enabled to describe services and explain to customers how the functionalities offered by such services can be consumed. Essentially, syntactic description is concern with using XML schemas and WSDL interface to specify the information model and functional capabilities respectively. While non-functional parameters are characterized using standard web services specifications like WS-policy, WS-agreement, WS-reliability, etc. [29].

The syntactic service description approach is lightweight that is, does not involve the use of semantic or ontology techniques, which are too computationally complex and resource intensive [10]. Therefore, in the context of AMC, syntactic approach to service description is most ideal. Nonetheless, WSDL standard suffers the limitation that it only operates at the syntactic level and lacks the semantic expressivity needed to unambiguously represent the requirements and capabilities of a web service [29], [30], [31].

This limitation can be summarized into two basic weaknesses of Syntactic web service description [30]:

- Lack of semantic supports that helps to indicate the meaning and semantic constraints of data involved in web services. This may lead to a high probability of generating ambiguities during service discovery.
- It does not cover the capabilities of a web service in the description. With this, there is the tendency that the matchmaking process will not be capable of recognizing the similarity between capabilities of a provided web service and the functionalities of the web service being requested. This weakness becomes obvious when there are multiple service providers offering web services with similar functionality but having dissimilar non-functional capabilities [32]. Moreover, it can be possible for two services to have the same syntactic definition but perform significantly different functions, while another two syntactically dissimilar services can perform the same function [30].

Several extensions to the WSDL standard such as SAWSDL, WS-Policy, and WSDL-M have been proposed to minimize the above highlighted weakness [1], [33].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.1.2 Semantic web service description

In contrast to non-semantic web service description, the semantic approach relies on ontologies to describe web services [31]. Ontology is a formal explicit specification of shared conceptualization [34].

The chief essence of semantic description is to enrich service description through enhanced expressivity. Such enhancements can benefit automation in service discovery and improve the efficiency of discovering relevant web services. Also, by semantically describing a service, it becomes possible for the service to be linked to abstract and cross-platform concepts. This sematic-enabled linking introduces robustness for example, facilitating the possibility to resolve and match different semantic representations and the mapping between services from different providers [27].

A key distinction between syntactic and semantic description approaches is that the latter's expressiveness allows for a broader perspective to service description. This expanded perspective adequately covers both functional and non-functional properties like availability, scalability, reliability etc.

With regards to service discovery, a foremost advantage of semantic description is the capability to allow for unambiguous web service specification, which addresses the lack of understanding of the semantic meaning of messages and data [27]. Using the semantic approach, a service is described in terms of profiles, models, and bindings. Information associated with the functionalities of the service is contained in the service profile. Service implementations, required input, and the expected output are described by the service model. The information model is defined through domain ontologies. Capabilities and functional categories are used to characterize functional details while non-functional parameters are described by the use of ontologies. Examples of semantic description languages include Web Ontology Language for Services (WOL-S), Semantic Web Services Ontology (SWSO), Web Service Modeling Ontology (WSMO), WSMO-Lite etc.

In comparison, the semantic approach achieves more elaborate web service descriptions than the syntactic counterpart. But despite the fact that semantic approaches augment web service description, using semantic techniques on current mobile devices is not a trivial task and computational wise, it is too resource-intensive [9], [35].

4.2 Web Service Implementation Frameworks

Web services achieve its goal by implementing the SOA paradigm of "software-as-a-service" (SaaS) in a technology-neutral fashion that is, independent of hardware platform, programming languages and even operating systems. Such technology independence is realized by providing well-defined interfaces for distributed functionalities.

The advantage of this technology neutrality is that the complexity with regards to heterogeneity that characterize the web services provisioning space is bridged; distributed functionalities otherwise services, which may be running on dissimilar hardware and software platforms can still communicate effectively through web service interfaces [36]. Interoperability in web services is achieved by using the Extensible Markup Language (XML) technology to define and implement message exchange or intercommunication protocols, which we refer to as implementation framework. This section explores the two predominant web service implementation frameworks:

4.2.1 The SOAP-based

Simple Object Access Protocol (SOAP) is a messaging framework for web services realization. The SOAP-based framework follows the Remote Procedure Call (RPC) style of web service interaction in which service providers and consumers are required to establish a common understanding with respect to service syntax and operations to facilitate inter communication.

At a fundamental level, the core idea in the SOAP-based framework revolves around the exchange of XML encoded messages over Hyper-Text Transmission Protocol (HTTP). This simply means that the SOAP protocol works by exchanging messages over HTTP using GET/POST operations [5]. SOAP is the traditional standard for web service implementation hence it is widely adopted because, among other reasons, it is supported by massive development tools [36], [37].

By configuration, SOAP framework is custom-made for fixed networks which make it heavy in nature. As described in [38], "SOAP is all about servers talking to servers, with rigid standards, extensive design, serious programming, and heavyweight infrastructure all essential parts of the equation".

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Furthermore, since the SOAP framework was originally designed for fixed networks, performance factors such as heavy payload, tight coupling, mobility etc., which now constitute huge resource burden to mobile devices, were not considered. Therefore, SOAP-based web services are not the best candidates for resource constrained environments.

4.2.2 The REST-based

The philosophy of the Representational State Transfer (REST) and its SOAP counterpart are very different. For example, while SOAP adheres to the RPC model, REST in the contrary, follows the concept that models web services as resources and focuses on using the intrinsic power of HTTP to retrieve representations of these resources in their varying states. That is, based on the REST framework, web services functionality are exposed as resources and signified by a distinctive Uniform Resource Identifier (URI). Then, a set of well-known, standard operations – GET, POST, PUT, and DELETE are used to operate on the resources [39].

Unlike SOAP, the REST messaging framework has lightweight payload which makes it suitable for mobile networks [37], [38], [40].

Although both REST and SOAP frameworks can be used to implement web services, each has its own strengths and weaknesses. In the next subsection we present selected key performance parameters that are used to compare both frameworks in line with the challenges of AMC as discussed in section 3.

4.2.3 Evaluation parameters of web service implementation framework

Web services are tending towards the mobile wireless world as an emerging technology for mobile environments. This inclination is driving research effort to focus on how mobile devices can operate both as web service providers and clients [41], [42].

An imperative requirement of this research direction is to provide uninterrupted, lightweight, web services to resource constrained devices operating in dynamic environments. Consequently, performance evaluation researches have used diverse parameters to evaluate SOAP and REST web service frameworks with regards to their suitability for resource-constrained environments.

Some of the key evaluation parameters that have greater implications on mobile devices' resources and quality of service include:

a. Payload: In SOA, payload refers to the actual message content that is exchanged between applications. SOAP and REST adopts different web services information model implementation therefore, their message sizes (payload) are equally different. Bearing in mind the concern of limited resources in AMC, any communication model with heavy payload may likely result in unacceptable performance overheads. Such performance overhead is mainly due to the encoding and decoding of messages, which contradicts the resource requirements of mobile devices.

b. Flexibility: In the general sense, flexibility is an attribute or design principle of a software that makes it possible for the software to adapt to different requirements. In the context of literature, flexibility is considered in terms of the ability to allow varying types of return data [37]. Because AMC is characterized by dynamic context and consist of heterogeneous mobile nodes, a flexible information model will be a more ideal choice. For example certain approaches may require only XML data format or binary attachment parsing, which makes adaptability difficult in environments with heterogeneous devices.

c. Response time: This parameter refers to the overall time it takes to get feedback for a service request. The effect of response time can be viewed from two perspectives: device resources and quality of service. Generally, response time is directly proportional to energy (battery) consumption [43]. And on the aspect of quality of service, long response time leads to clients' dissatisfaction. Therefore, a good web service development framework for AMC should guarantee the design of web services with short response time to be able to achieve optimal battery consumption.

d. Energy consumption: As pointed out in [44], energy consumption (the rate of battery depletion) is one of the major challenges being faced by mobile devices, especially mobile phones. Several factors are responsible for the power consumption rate of web services. Such determinants includes, amongst others, response time, payload or message size, bandwidth requirement etc. Interestingly, these factors that influence power consumption rate of a web service are characteristic of the development framework. Since AMC relies on mobile nodes that are battery dependent, it is imperative to take into consideration the power

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

consumption rate when choosing a development framework.

e. Memory footprint: The amount of memory required to process a web service is called memory footprint. With mobile nodes, memory or processing capability is a scares resource. And in AMC, the principal operation is about sending requests and receiving responses, which require sufficient memory. To avoid undue resource burden, AMC requires web services with minimal memory footprint.

f. Caching: Caching is a useful technique for boosting performance in web service provisioning by minimizing response time. The technique allows pre-fetch data to be temporally stored to service future requests for such data faster. Mobile devices benefit from catching because is reducing processing time especially in instances where data is accessed remotely.

g. Bandwidth requirement: The rate at which data can be transferred between devices via wired or wireless media within a specific time is referred to bandwidth. This data transmission rate requires resources. For instance, heavy bandwidth drains battery because it requires more memory and processing time. Different web service frameworks have varying data transmission rates. However, to achieve optimal performance in AMC, it is highly desirable to mitigated resource burden by adopted a framework with light bandwidth.

h. Scalability: Scalability is described in [45] as the ability to handle growth efficiently. This parameter is considered as one of the vital factors that determines the suitability of a web service framework for mobile environments [39]. The fact that in AMC devices are free to leave or join the network at any time makes the issue of scalability vital because at some point the volume of interaction may grow as more devices joint the Ad-hoc Cloud.

i. Coupling: The manner and degree of interdependence between software modules is termed coupling in SOA. Coupling is used to judge how closely connected two modules are, and the strength of the relationships between modules [44]. System modules are either tightly or loosely coupled. Tight coupling reduces module independence thereby posing the need for a central point of control, which is against the principles of ad-hoc computing.

j. Mobility: Mobility is the central phenomenon in Ad-hoc Mobile Cloud, were mobile nodes are free to join and leave the network at will. For such operational flexibility to be achieved in mobile web services provisioning, AMC environment must integrate web services standards with support for mobility.

4.2.4 Comparative evaluation of SOAP and REST

Based on the above evaluation parameters we compare SOAP and REST frameworks to determine their suitability for use in AMC. For the sake of clarity, the selected parameters for comparison are further categorized into two groups: design flexibility and performance overheads as shown Table 1. Outside these parameters used in our comparison, each framework has its own distinctive features and shortcomings that make it more or less suitable for certain types of application environment.

5. MOBILE WEB SERVICE ARCHITECTURES

The recent decades have witnessed increased research interest in mobile web services. These research efforts spans across the entire scope of realizing a web service provisioning platform that is independent of the conventional Cloud infrastructure [6], [46], [47], [48], [49].

Towards the above objective, considerable research effort has been directed on formulating an architectural framework for mobile web services. Although still in their infancy, literature reveals several proposed architectures as summarized in Figure 3. It is also evident that each of these architectures tries to address a particular issue related to the challenges faced by mobile devices [5].

Figure 3 illustrates the four notable types of AMC web service architectures as classified by [5], [22].

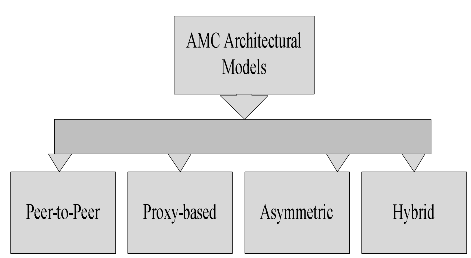


Figure 3: AMC Architectural Models

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: Comparison of SOAP and REST Web Services Frameworks

Evaluation Parameters		SOAP Framework	REST Framework
Design style	Flexibility	Not flexible: * Returns only XML data * Requires binary attachment parsing	Flexible: * Supports all data types * Returns all data types
	Caching	Not supported	Supported
	Coupling	Tightly coupled	Loosely coupled
	Scalability	Less scalable	More scalable
	Mobility	Not supported	Supported
Overheads	Energy consumption	High	Low
	Bandwidth requirement	High	Low
	Response time	Long	Short
	Payload	Heavy	Light
	Memory footprint	Large	Small
Suitability for Ad-hoc Mobile Cloud			
Evaluation parameters		SOAP	REST
Flexibility		×	√
Caching		×	√
Coupling		×	√
Scalability		√	√√
Mobility		×	√
Energy consumption		√	√√
Bandwidth		√	√√
Response time		√	√√
Payload		√	√√
Memory footprint		√	√√

5.1 Peer-to-Peer-based AMC Architecture

The infrastructure-less nature of AMC forms the rationale for the Peer-to-Peer architectural model. In fact, the AMC paradigm is based on the Peer-to-Peer philosophy. For example, Peer-to-Peer is a computing model that enables efficient, rich, and low-cost resource sharing in a distributed and collaborative manner, which perfectly defines AMC with respect to the web service domain. Peer-to-peer AMC architecture as shown in Figure 4, consist of at least two mobile nodes of equal responsibility. That is, each node is capable of acting as service provider or client by utilizing the advertising mechanism of peer-to-peer networks for web service publishing and discovery [22].

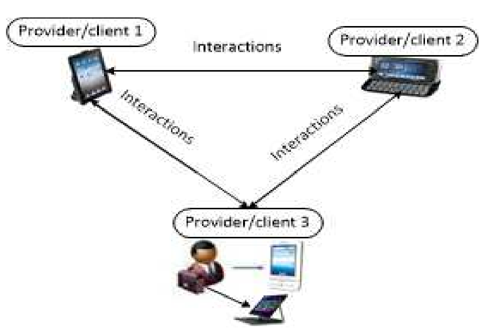


Figure 4: Peer-to-Peer AMC Architecture

This advertisement approach uses the Lifetime Concept, which makes it possible to deal with node mobility and the dynamic binding of web service information in WSDL documents.

The lifetime concept is basically deployed as a strategy to avoid resource burden created by maintaining up-to-date web service descriptions in a centralized registry. To achieve this goal, a lifetime parameter is used to set the validity period of service advertisements broadcasted by active devices in the network. At the expiration of the lifetime, the corresponding advertisement is deleted or marked invalid. This approach helps to compensate for the constraints of mobile devices, which are the key components of the Ad-hoc Mobile Cloud.

Technology wise, early implementations of the peer-to-peer AMC architecture are based on JXTA open source technology [50]. The JXTA platform support web service publishing and discovery via JXTA protocols, which incorporates WSDL documents into Model Specification Advertisements (MSA). With this, a web service is recognized as a JXTA service among providing peers [51]. There is also a protocol for peer identification, which ensures that only certified peer devices are allowed to join the network by assigning unique PeerID to every participating node.

Once a network of peers is enabled, communication among peers occurs via a channel

Journal of Theoretical and Applied Information Technology
15th August 2016. Vol.90. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

while participating nodes listen to the channel to receive messages [52].

5.2 Proxy-based AMC Architecture

Proponents of the proxy architecture emphasize the need to evade the numerous challenges associated with services provisioning in AMC such as scalability, protocol compatibility and processing power. As illustrated in Figure 5, in proxy-based architecture, web services are hosted by mobile devices but served through a high-end node (proxy) attached to a fixed network. So, there are basically two components in the proxy-based architectural setting: 1) A mobile service provider (the actual host) and 2) a more powerful machine that acts as an intermediary between mobile hosts to perform resource-intensive processes. This intermediary is often called a proxy or surrogate. Although the proxy is only a high-end intermediary, virtually, it appears to the client device as the actual service provider.

With the proxy-based approach, catching operations can be offered to supported intermittent connectivity.

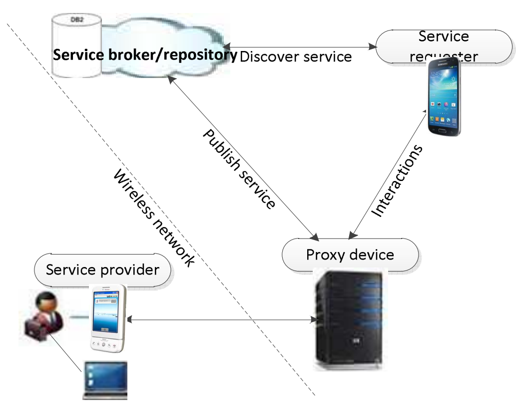


Figure 5: Proxy AMC Architecture

Other benefits include the capability to complement limited bandwidth of mobile devices since the proxy is often attached to the fixed network and enhance scalability and higher performance in terms of response time.

The aforementioned benefits of the proxy-based architecture have some advantages over the other architectures with respect to suitability for AMC service provisioning. However, it also introduces the challenge of synchronization between the real mobile host and the proxy, which is only a representation of the real mobile host [22]. That is any change made to a web service from the host's side must simultaneously reflect on the proxy. Another problem with proxy-based architecture is that it constitutes a single point of failure because the core processes of web service publishing and discovery are tired to a single proxy server.

Proxy-based architecture relies on the Jini technology, which is a Java-based development infrastructure [53]. The Jini infrastructure consists of two major modules: lookup service and join/discovery protocol. Lookup service is the serves as the service repository while the join/discovery protocol is used for publishing and discovering services. In Jini, services are published by the proxy to a lookup registry from where client can discover services [53]. When a potential consumer locates a service on the lookup registry, it retrieves its binding information and then contacts the proxy to invoke the service.

5.3 Asymmetric-based AMC Architecture

The asymmetric AMC web service provisioning architecture was first propose in the work of [54]. This architectural model follows the general web services architecture, where service descriptions are published in a UDDI registry. Unlike in general web services architecture, asymmetric style tries to avoid the performance complexities associated with extracting some complex SOAP types [54]. To realize this, only simple SOAP types such as String, Integer, and Char are accepted.

As depicted in Figure 6, asymmetric architecture does not require a proxy instead the service provider is an actual mobile node.

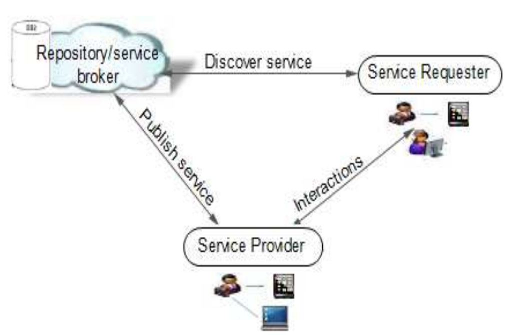


Figure 6: Asymmetric AMC Architecture

This restriction on SOAP types is meant to address the challenge of resource limitation in mobile devices. As used in the work of [55], the term "Asymmetric" refers to architectures that are designed with the consideration of resource constraints in mobile nodes. Another distinctive aspect of asymmetric architecture is the use of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Asynchronous Service Access Protocol (ASAP) [56] to implement web services interactions.

ASAP allows asynchronous and independent execution of called web services. That is to say, a client can invoke a service and then wait for its response without necessarily blocking the device in the course of execution [5].

The major asynchronous interactions supported by ASAP are the callback and polling operations, which enable the server device to return the response to a service request whenever it is ready and allow client to re-connect to check whether the results of a requested service are ready respectively. Although asymmetric architecture has interesting features, its support for only SOAP types can limit service capabilities in addition to constituting a resource burden generated by SOAP overheads [22].

5.4 Hybrid AMC Architecture

Theoretically, the hybrid architecture has been proposed by [5] to possibly compensate for the individual weaknesses of the three architectures. The hybrid architecture consists of all the three architectures coexisting independently. An ideal technology for hybrid architecture can a Peer-to-Peer overlay network because it offers good and powerful templates for creating extensive data sharing, content distribution and application-level multicast applications [57].

5.5 Comparison of AMC Architectural Models

A comparison of the three architectural styles with respect to the challenges of AMC is rendered in this section as depicted in Table 2. The comparative analysis is based on five evaluation parameters namely, architectural style, scalability, discovery approach, resource-intensiveness, and mobility support. The comparison is not intended to probe the technical or operational capabilities of these architectures but rather to emphasize on the features that should inform the choice of suitable architecture for AMC.

Table 2: Comparison of Mobile Web Service Architectures

Evaluation Parameter	Peer-to-Peer	Proxy-based	Asymmetric
Arch. style	Decentalized	Centralized	Cetralized
Scalability	More scalable	Scalable	Limited
Discovery approach	P2P advertisement	UDDI-like	UDDI-like
Resource-intensiveness	No	No	Yes
Mobility support	Supported	Limited support	Limited support
Suitability for Ad-hoc Mobile Cloud			
Evaluation parameter	Peer-to-Peer	Proxy-based	Asymmetric
Arch. style	√√	√	√
Scalability	√√	√	√
Discovery approach	√√	×	×
Resource-intensiveness	√√	√	×
Mobility support	√√	√	√

6. STATE-OF-THE-ART IN AMC SERVICE DISCOVERY

Current research is paving way for mobile web service provisioning as a major step to realize pervasive computing and expand revenue generation [5], [58]. These research efforts are primarily motivated by the need to ease the various challenges of accessing remote web services via mobile devices.

In the above direction, recent years has witness increased research activities geared towards evolving discovery mechanisms that take into account the unique characteristics of mobile environments, especially in terms of resource constraints and dynamic context.

Based on existing literature, we identify, discuss and classify existing works into three schools of taught: i) using light-weight semantic techniques,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ii) offloading computational-intensive task to the Cloud, and iii) the pure ad-hoc approach.

6.1 Use of Lightweight Semantic Techniques

Generally, web service discovery mechanisms in Cloud, Mobile Cloud, and other large-scale service oriented domains are predominantly enhanced via semantic and ontology techniques [59]. Although such enhancements improve service discovery efficiency, and accuracy, they are most suitable in the aforementioned domains where computational resource is not a challenge [60].

In order to extend the benefits of semantics techniques to mobile web service discovery, scholars have proposed the use of optimized technologies to cater for the unique limitations of mobile devices.

Within this same school of taught, there are two areas of emphasis. The first dimension focuses on enriching service description using extensions of WSDL as way of improving service discovery. These extensions such as Semantic Annotation Web Service Description Language (SAWDL) and Web Service Policy (WS-Policy) basically enhance service descriptions by providing interfaces (e.g. ModelReference, liftingSchemaMapping and loweringSchemaMapping) that help to link or annotate WSDL elements to semantic concepts.

Providing mechanisms to link WSDL elements to semantic concepts removes the requirement to formally specify semantic models e.g. ontology.

The other area of emphasis attempts to deal with improving service discovery by enhancing the web service matchmaking process using optimized or lightweight semantic reasoners. Currently, there are a number of such optimized tools purposely designed to reduce the computational-intensiveness associated with traditional semantic matchmakers. Examples of these include: mTableaux [10], WSMO-lite [61], Delta Reasoner [62], and MobiDisc [63] etc.

6.2 Offloading of Computation to the Cloud

Resource constraint has been identified as one of the primary challenges facing service discovery in mobile environments. This challenge is principally due to their inherent nature. Consequently, to bridge the gap between resource-constrained environments and resource-intensive web service discovery, the second school of taught explores the option of pushing resource-intensive tasks to the resource-rich Cloud. The goal then, is to develop service discovery mechanisms that can achieve efficient service discovery with minimized resource burden. This approach incorporates an intelligent decision making agent (algorithm) that decides when and how to initiate a process of hand-over or offload of computation from the mobile node to the Cloud. Apparently, with this setup, the potentials of semantic service discovery can be harnessed in delivering efficient and accurate web service discovery in resource-constrained environments. Some notable efforts in this direction are reported in [11], [44], [11], [64].

6.3 Pure Ad-hoc Approach

Owing to the infrastructure-less nature of MAC, the last category of scholars advocates a decentralized approach to AMC service discovery. That is, utilizing a pure ad-hoc framework where the web service discovery process takes place between mobile peers without the involvement of any high-end or Cloud server. This category adopts resource-friendly (non-semantic) techniques for web service description like WSDL-M [1]. Also, to make up for the semantic deficiency, various relevancy algorithms have been utilized here to optimize service discovery [1], [65], [66].

An early work in this domain with a Peer-to-Peer mechanism for publishing and discovering web services based JXTA technology was reported in [45]. Current research efforts such as [6], [67] have leveraged on advances in mobile device's capabilities to develop service discovery mechanisms based on nascent technologies like OSGi, WiFi, and Wi-Fi Direct.

6.4 Investigative Evaluation of AMC Service Discovery Approaches

This section investigates some notable research efforts within the aforementioned schools of taught in AMC service discovery. The purpose of this investigation is to evaluate them based on selected parameters that are chosen based on their influence on the three major challenges facing service discovery in AMC. These parameters are:

a. Matchmaking Technique: this process can impact on resource availability. There are four types of matchmaking techniques according to [66]: keyword – resource-efficient but only ineffective with large result sets; Syntactic-based – combines keyword and an algorithm that filters services based functionality, operation, etc.; pragmatic technique can be any of the above techniques or a hybrid. It incorporates context parameters into the matchmaking process; Semantic – uses semantic tools such WSMO, WSDL-S etc.

Journal of Theoretical and Applied Information Technology
15th August 2016. Vol.90. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

b. Context awareness: handling the dynamic context in AMC requires the use of context information gathered via context ware techniques. Our evaluation focuses on three context types: User context (UC) e.g. location, preferences ratings etc.; Device context (DC), consisting of resource context and (RC) device profile (DP); and Service context (SC) also called Quality of Web Service.

c. Proactive discovery: discovering relevant services in AMC needs a form of proactive response so that the discovered services can be tailored to the current context clients.

d. Relevancy ranking: the lack of semantic interpretation of services increases the chances of ambiguity in service discovery. On the other hand, using non-semantic approaches require additional techniques to determine service relevance. Relevancy ranking has become a common context-aware service discovery technique [1], [67].

Based on these parameters, an evaluation of prominent works in this domain is offered in Table 3.

Table 3: Investigative Evaluation of AMC Discovery Approaches

Discovery category		Matchmaking technique				Context-awareness				Proactive discovery	Relevancy ranking
		Syntactic	Semantic	Keyward	Pragmatic	UC	DC	SC (QoWS)			
								RC	DP		
Lightweight technology	mTableauxv'09 [10]	×	√	×	×	×	×	×	×	×	×
	Delta reasoner '12: [62]	×	√	×	×	√	×	√	×	×	×
	WSMO-lite '13: [61]	×	√	×	×	√	×	√	√	×	×
	MobiDisc '15: [63]	×	√	×	×	×	×	×	×	×	×
Offloading computation to Cloud	Elgazzar et. al, '13: [65]	×	√	×	√	×	√	×	×	×	×
	Ravi et. al, '13: [45]	–	–	–	–	×	×	√	√	×	×
	CMDis '14: [64]	×	×	√	×	×	×	√	×	×	√
	DaaS '14: [11]	√	√	√	√	×	×	√	×	×	√
Pure ad-hoc approach	MobiEureka "10: [1]	√	×	√	×	×	×	√	×	×	×
	Srirama "13: [6]	√	√	√	×	√	×	×	×	×	×
	Elgazzar et. al, '13: [66]	×	×	√	√	√	×	√	×	×	√
	Al-Masri '09: [67]	√	×	√	×	×	×	√	×	×	√

7. CONCLUSION

The rapid advancement in mobile devices' capabilities and mobile communication technologies means that AMC has a promising future. Furthermore, the proliferation of Cloud e-markets has generated a corresponding growth in consumer base of mobile services [58].

In order to gain insight into current trends and the future of AMC, we have compared web service description approaches, implementation frameworks, and mobile web service architectures while focusing on how different web services standards, techniques or approaches impact on the AMC service discovery process. The comparison is motivated by the need to recognize the requirements for achieving effective AMC service discovery. We now conclude that semantic tools are less suitable for AMC than non-semantic approaches. Furthermore, the REST framework is shown to be more ideal for AMC than SOAP.

Concerning architectural models for mobile services, the decentralized model (P2P) is the most appropriate because, among other benefits, it matches the default nature of AMC.

We have also explored the state-of-the-art in AMC with a view to determining strategies that have been employed to enhance service discovery efficiency by tackling the central challenges in that domain. From this we have come to the following conclusions:

a. Although context awareness is a fundamental phenomenon in mobile web service discovery

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and is being used in almost all the selected approaches, some vital components of context information are still being employed sparingly by current AMC service discovery mechanisms. For instance, resource context, defined as the state of a device's resources (battery, memory) [57] and QoWS are the most underutilized context parameters, based on our evaluation. Future discovery systems should take these aspects of context into account.

b. The combination of an efficient matchmaking process and useful context information may not guarantee the discovery of relevant web services in a dynamic environment when the discovery process lacks proactive capabilities. Designing service discovery mechanisms with proactive capability will be a future requirement for AMC.

Our arguments are drawn from the observation summarized in Table 3, that none of the selected approaches incorporated proactive techniques like service request or resource adaptation in the discovery process. Our analysis has established the need to:

- Develop proactive mobile service discovery mechanisms as a potent future direction to address the issue of dynamic context change in AMC;
- Expand the scope of device context needs to include resource context in order to consider service relevance as a function of meeting both client requirements and device resource capabilities. This will enhance service discovery and facilitate support for resource-ware service discovery in the future;
- Develop mobility management algorithms in order to actualize the commercialization of AMC.

Implementing the findings of this paper within the context of Ad-hoc Mobile Cloud, are subject to the inherent challenges of mobile environments as well as the availability of suitable technology. However, the future study from this work will focus of implementing a lightweight, context-aware and proactive service discovery mechanism for AMC.

REFRENCES:

[1] E. Al-Masri and Q. H. Mahmoud, "MobiEureka: An approach for enhancing the discovery of mobile web services," *Pers. Ubiquitous Comput.*, vol. 14, no. 7, pp. 609–620, 2010.

[2] K. S. Wagh and R. Thool, "Web Service Provisioning on Android Mobile Host," *Int. J. Comput. Appl.*, vol. 81, no. 14, pp. 1–7, 2013.

[3] R. Lacuesta, J. Lloret, S. Sendra, and L. Peñalver, "Spontaneous Ad Hoc Mobile Cloud Computing Network," *Sci. World J.*, vol. 19, 2014.

[4] S. El-Masri and B. Suleiman, "A Framework for providing mobile web services," *Second Int. Conf. Innov. Inf. Technol.*, 2005.

[5] K. Elgazzar, H. Hassanein, and P. Martin, "Mobile Web Services : State of the Art and Challenges," *Int. J. Adv. Comput. Sci. Appl.*, vol. 5, no. 3, 2014.

[6] S. Srirama and C. Paniagua, "Mobile web service provisioning and discovery in android days," *Conf. Mob. Serv.*, 2013.

[7] H. Dai, X. (Robert) Luo, Q. Liao, and M. Cao, "Explaining consumer satisfaction of services: The role of innovativeness and emotion in an electronic mediated environment," *Decis. Support Syst.*, vol. 70, pp. 97–106, Feb. 2015.

[8] M. Miraoui, C. Tadj, J. Fattahi, and C. Ben Amar, "Dynamic Context-Aware and Limited Resources-Aware Service Adaptation for Pervasive Computing," *Adv. Softw. Eng.*, vol. 2011, pp. 1–11, 2011.

[9] K. Elgazzar, "Discovery , Personalization and Resource Provisioning of Mobile Services, Ph.D Thesis," *Queen's Univ. Kingston, Ontario, Canada*, no. August, 2013.

[10] L. A. Steller, "Light-weight and Adaptive reasoning for mobile web services, Ph.D thesis, Monash University, Australia." 2010.

[11] K. Elgazzar, H. Hassanein, and P. Martin, "DaaS: Cloud-based Mobile Web service Discovery," *Pervasive Mob. Comput.*, vol. 13, pp. 67–84, 2014.

[12] K. Dejan, Y. Cao, and R. Klamma, "Mobile Cloud computing: A comparison of application models," *arXiv Prepr. arXiv1107.4940*, 2011.

[13] N. Palmer, R. Kemp, and T. Kielmann, "Ibis for mobility," in *Proceedings of the 10th workshop on Mobile Computing Systems and Applications - HotMobile '09*, 2009, pp. 1–6.

[14] M. Satyanarayanan, "Pervasive computing: vision and challenges," *IEEE Pers. Commun.*, vol. 8, no. 4, pp. 10–17, 2001.

[15] M. Satyanarayanan and P. Bahl, "The case for vm-based cloudlets in mobile computing," *Pervasive Comput.*, 2009.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[16] S. K. Saha, V. Rudakova, and F. A. Cheikh, "Collaborative cameras for multi-target tracking using color based particle filter and object contour," in *3rd European Workshop on Visual Information Processing*, 2011, pp. 138–142.

[17] U. Bellur, H. Vadodaria, and A. Gupta, "Semantic Matchmaking Algorithms," no. November, 2008.

[19] Y.-S. Chen, C.-S. Hsu, and C.-H. Cheng, "Network mobility protocol for vehicular ad hoc networks," *Int. J. Commun. Syst.*, p. n/a–n/a, Mar. 2013.

[20] N. Fernando, S. Loke, and W. Rahayu, "Mobile Cloud computing: A survey," *Futur. Gener. Comput. Syst.*, vol. 29, no. 1, pp. 84–106, 2013.

[21] P. Papakos, C. Licia, and D. S. Rosenblum, "VOLARE: Context-Aware Adaptive Cloud Service Discovery for Mobile Systems," in *Proceedings of the 9th International Workshop on Adaptive and Reflective Middleware*, 2010, pp. 32–38.

[22] C. Doulkeridis and M. Vazirgiannis, "CASD: Management of a context-aware service directory," *Pervasive Mob. Comput.*, 2008.

[23] P. Pawar and S. Srirama, "A comparative study of nomadic mobile service provisioning approaches," *Int. Conf. Next Gener. Mob. Appl. Serv. Technol. (NGMAST' 07)*, 2007.

[24] Seokung Yoon, SoonTai Park, and Haeryoung Park, "Security Analysis of Vulnerable Wi-Fi Direct," *Computing and Networking Technology (ICCNT), 2012 8th International Conference on*. pp. 340–343, 2012.

[25] J. S. Rellermeyer and S. Bagchi, "Dependability as a Cloud service - a modular approach," in *IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN 2012)*, 2012, pp. 1–6.

[26] IBM, "Understanding web service interoperability." 01-Feb-2014.

[27] D. Fensel, F. Fischer, and J. Kopecký, "WSMO-Lite: Lightweight Semantic Descriptions for Services on the Web," 2010. [Online]. Available: http://www.w3.org/Submission/WSMO-Lite/. [Accessed: 10-Jun-2015].

[28] B. Di Martino, G. Cretella, and A. Esposito, "Semantic Representation of Cloud Services: A Case Study for Microsoft Windows Azure," in *2014 International Conference on Intelligent Networking and Collaborative Systems*, 2014, pp. 647–652.

[29] C. Erik, C. Francisco, and M. Greg, "Web Service Definition Language (WSDL)," *W3C Technical note*, 2001. [Online]. Available: http://www.w3.org/TR/wsdl. [Accessed: 09-Jun-2015].

[30] H. Dong, F. Hussain, and E. Chang, "Semantic Web Service matchmakers: state of the art and challenges," *Concurr. Comput. Pract. Exp.*, no. 25, pp. 961–988, 2013.

[31] R. Akkiraju, J. Farrell, and J. Miller, "Web Service Semantics - WSDL-S," *Kno.e.sis Publications*. 2005.

[32] H. Omrana, I. El Bitar, and F.-Z. Belouadha, "A Comparative Evaluation of Web Services Description Approaches," in *2013 10th International Conference on Information Technology: New Generations*, 2013, pp. 60–64.

[33] R. Phalnikar and P. A. Khutade, "Survey of QoS based web service discovery," *2012 World Congr. Inf. Commun. Technol.*, pp. 657–661, Oct. 2012.

[34] A. Vedamuthu, D. Orchard, and F. Hirsch, "Web services policy 1.5-framework," *W3C ...*, 2007.

[35] D. Martin, M. Burstein, and J. Hobbs, "OWL-S: Semantic markup for web services," *W3C Memb. Submiss.*, 2004.

[36] S. Abolfazli, Z. Sanaei, A. Gani, F. Xia, and L. T. Yang, "Rich Mobile Applications: Genesis, taxonomy, and open issues," *J. Netw. Comput. Appl.*, vol. 40, pp. 345–362, Apr. 2014.

[37] P. Castillo, J. Bernier, and M. Arenas, "SOAP vs REST: Comparing a master-slave GA implementation," *arXiv Prepr. arXiv*, 2011.

[38] K. Wagh and R. Thool, "A Comparative Study of SOAP Vs REST Web Services Provisioning Techniques for Mobile Host," *Journal of Information Engineering and Applications*, vol. 2, no. 5. pp. 12–16, 2012.

[39] F. AlShahwan and K. Moessner, "Providing SOAP Web Services and RESTful Web Services from Mobile Hosts," in *2010 Fifth International Conference on Internet and Web Applications and Services*, 2010, pp. 174–179.

[40] G. Mulligan and D. Gracanin, "A comparison of SOAP & REST implementations of a service based interaction independence middleware framework," in *Pro. of the 09 Winter*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Simulation Con. (WSC), 2009, pp. 1423–1432.

[41] L. Pham and G. Gehlen, "Wireless Conference 2005 - Next Gen. Wireless and Mobile Communications and Services (European Wireless), 11th European," *Wireless Con. 2005-Next Generation Wireless and Mobile Communications and Services (European Wireless), 11th European*. pp. 1–7, 2005.

[42] L. P. Guido Gehlen, "Mobile Web Services for Peer-to-Peer Applications," in *Second IEEE Consumer Communications and Networking Con., 2005. CCNC. 2005*, 2005, pp. 427–433.

[43] Y. Cao, M. Jarke, and R. Klamma, "Mobile Access to MPEG-7 Based Multimedia Services," *2009 Tenth Int. Conf. Mob. Data Manag. Syst. Serv. Middlew.*, pp. 102–111, 2009.

[44] H. Hamad, M. Saad, and R. Abed, "Performance Evaluation of RESTful Web Services for Mobile Devices," *Int. Arab J. e-Technology*, vol. 1, no. 3, pp. 72–78, 2010.

[45] A. Ravi and S. K. Peddoju, "Energy Efficient Seamless Service Provisioning in Mobile Cloud Computing," *2013 IEEE Seventh Int. Symp. Serv. Syst. Eng.*, pp. 463–471, Mar. 2013.

[46] W3C, "Web Service Scalability and Performance with Optimising Intermediaries," 2001. [Online]. Available: http://www.w3.org/2001/03/WSWS-popa/paper33. [Accessed: 16-Jun-2015].

[47] R. Want, T. Pering, and G. Danneels, "The Personal Server: Changing the way we Think about Ubiquitous Computing," *Int. Conf. Ubiquitous Comput.*, pp. 194–209, 2002.

[48] S. N. Srirama, M. Jarke, and H. Zhu, "Scalable Mobile Web Service Discovery in Peer to Peer Networks," in *2008 Third International Conference on Internet and Web Applications and Services*, 2008, pp. 668–674.

[49] Y. S. Kim and K. H. Lee, "A light-weight framework for hosting web services on mobile devices," *Proc. 5th IEEE Eur. Conf. Web Serv. ECOWS 07*, pp. 255–263, 2007.

[50] K. Elgazzar, H. S. Hassanein, and P. Martin, "Personal Web Services : Architecture and Design," 2012.

[51] N. Maibaum and T. Mundt, "JXTA: a technology facilitating mobile peer-to-peer networks," in *International Mobility and Wireless Access Workshop*, 2002, pp. 7–13.

[52] S. N. Srirama, "Mobile Hosts in Enterprise Service Integration," *M.Sc Thesis, RWTH Aachen Univ. Ger.*, 2008.

[53] W. Yeager and J. Williams, "Secure peer-to-peer networking: the JXTA example," *IT Prof.*, vol. 4, no. 2, pp. 53–57, Mar. 2002.

[54] A. Mian, R. Baldoni, and R. Beraldi, "A survey of service discovery protocols in multihop mobile ad hoc networks," *Pervasive Comput. IEEE*, 2009.

[55] I. Pratistha, N. Nicoloudis, and S. Cuce, "A Micro-Services Framework on Mobile Devices.," *ICWS*, 2003.

[56] T. La Porta, K. Sabnani, and R. Gitlin, "Challenges for nomadic computing: Mobility management and wireless communications," *Mob. Networks Appl.*, 1996

[57] M. Kushwaha, I. Amundson, X. Koutsoukos, S. Neema, and J. Sztipanovits, "OASiS: A Programming Framework for Service-Oriented Sensor Networks," in *2007 2nd International Conference on Communication Systems Software and Middleware*, 2007, pp. 1–8.

[58] E. Lua and J. Crowcroft, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Commun. Mag. Surv. Tutorials*, vol. 7, no. 2, 2005.

[59] N. A. Saadon and R. Mohamad, "A Comparative Evaluation of Web Service Discovery Approaches for Mobile Computing," *Int. Conf. Informatics Eng. Inf. Sci.*, pp. 238–252, 2011.

[60] G. Kousiouris, D. Kyriazis, T. Varvarigou, E. Oliveros, and P. Mandic, *Achieving Real-Time in Distributed Computing*. IGI Global, 2012.

[61] N. Saadon and R. Mohamad, "WSMO-lite based web service discovery algorithm for mobile environment," *Int. J. Adv. Soft Compu. Appl*, 2013.

[62] B. Motik, I. Horrocks, and S. Kim, "Delta-reasoner: a semantic web reasoner for an intelligent mobile platform," *Proc. 21st Int. onference World Wide Web*, pp. 63–72, 2012.

[63] B. Cheyma, G. Chirine, and B. R. Lotfi, *MobiDisc: Semantic Web Service Discovery Approach in Mobile Environments*, vol. 8954. Cham: Springer International Publishing.

[64] N. A. Saadon and R. Mohamad, "Cloud-based Mobile Web Service Discovery framework with semantic matchmaking approach," in *2014 8th. Malaysian Software Engineering Conference (MySEC)*, 2014, pp. 113–118.

[65] K. Elgazzar, P. Martin, and H. S. Hassanein, "Empowering mobile service

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

provisioning through Cloud assistance," *Proc. - 2013 IEEE/ACM 6th Int. Conf. Util. Cloud Comput. UCC 2013*, pp. 9–16, 2013.

[66] K. Elgazzar, P. Martin, and H. S. Hassanein, "Personalized Mobile Web Service Discovery," in *2013 IEEE Ninth World Congress on Services*, 2013, pp. 170–174.

[67] E. Al-Masri and Q. H. Mahmoud, "Device-Aware Discovery and Ranking of Mobile Services," in *2009 6th IEEE Consumer Communications and Networking Conference*, 2009, pp. 1–5.