MOBILE WEB SERVICE DISCOVERY: A CLOUDLET MEDIATED FRAMEWORK

Murimo Bethel Mutanga

*Department of ICT, Mangosuthu University of
Technology, South Africa
[\[mutangamb@mut.ac.za\]](mailto:mutangamb@mut.ac.za)*

Abstract

The mobile web services paradigm (MWS) promises anywhere and all-time availability of services by leveraging the proliferation of smart devices with advanced capabilities. Consequently, research is focusing on ad-hoc mobile cloud, which extends the functional and architectural scope of MWS to include platforms for offering inexpensive and collaborative services co-hosted and discovered among peer devices. Existing solutions to the service discovery challenge posed by such dynamic environments predominantly focus on algorithms that adopt context-based lightweight service discovery mechanisms. But such a focus pays less attention to the architectural perspective of the service discovery problem. This paper contextualized and discussed the emerging context and complexity of the MWS discovery challenge under two key subjects: operational effectiveness and functional efficiency, which then foregrounds the proposed MWS discovery framework. The framework is analyzed based on hypothetical usage scenarios, demonstrating a robust architectural approach to MWS discovery.

Key Words - Cloudlet; context-awareness; device context; framework; relevant services; service discovery.

1. Introduction

Web services have continued to soar in popularity in the last decade, terms of the scale of their adoption and scope of application [1], [2]. As a result, the Web services concept now fundamentally defines the core of global business models because of its associated benefits of flexibility and cost-effectiveness, among other advantages. Apart from helping to cut the operating cost of big business corporations, these two benefits are also hugely responsible for improving and strengthening the viability of Small, Medium, and Micro Enterprise Businesses (SMMEs) due to the ease of creating business diversity [3], [4] introduced by Web services. Web services have revolutionized the e-marketplace and empowered global business enterprises. For instance, through web services, businesses can now readily enhance their capacity to build and use tailored business intelligence to scale up the quality of service delivery [5]. Essentially, the concept of Web services enables mundane, critical, or complex business processes (services) to be automated or encoded as pieces of software that makes itself available over the internet for use or invocation via a standardized XML messaging system [6]. Now, as the synergy between traditional and mobile cloud computing grows more robust, the interaction space for MWS delivery has correspondingly increased. This trend, among other factors, is driven by the fast and innovative advances experienced in ubiquitous computing in general, as characterized by a

surge in the number of high-capability smart devices and affordable and accessible supporting technologies [1], [7].

For example, today, over 80% of the world's population had access to various smart devices that are interestingly more advanced in functionalities than the traditional mobile phone used to store media to tools for accessing media from other devices. Both in terms of capability and functionality, smart devices are getting increasingly powerful such that they can now accomplish more advanced tasks such as accessing applications and value-added services offered remotely [8], [9] [10], [9]. Moreover, while one of the most crucial challenges facing mobile Web service provisioning, in general, has been the inability to run or at least the resource implications of running semantic-based service discovery algorithms on resource-constrained devices [9], [11]. However, in a scholarly study, Bobed et al. [12] interestingly demonstrated the feasibility of running Description Logics (DL) reasoners on Android-based smart devices. Based on a performance analysis of 300 ontologies from the ORE 2013 ontology set on current smartphones and tablets, the authors concluded that "using semantic APIs and reasoners on current mobile devices is feasible". This conclusion, therefore, further research aimed at broadening the scope of mobile web service offerings becomes highly imperative. Concerning supporting technologies, the last decade has seen a dramatic rise in various technologies that currently support or will further enhance the dominance of role MWS in the global e-marketplace. Among these state-of-the-art technologies. These technologies, which can

potentially transform and redefine the mobile cloud market, include but are not limited to Cloudlets, the emerging 5G networks [13] HTML5, Hypervisor [14], and Web 4.0 [15]. Specifically, these technologies strengthen capabilities such as facilitating the specifications for offline support, enabling an architecture-neutral execution of Web services on any smartphone and guaranteeing efficiency in Web service provisioning and discovery in the context of reducing latency and increasing response time [16], [17].

Furthermore, with the emergence and fast development of the Internet of Things (IoT), the drive to realize more innovative and unlimited interconnected interaction between people, devices, and even things, has become more practicable [18]. In the context of this study, the implication is that these IoT-enabled devices, in combination with advanced mobile communication technologies, can facilitate the design of more service provisioning frameworks that can integrate billions of cloud-based mobile sensors and wearable computing devices [19]. In all, the advances highlighted above indicate that the MWS-driven e-marketplace will continue to grow, which, therefore, underscores the need to explore the possibility of expanding the scope of mobile Web service provisioning beyond the traditional mobile cloud setting. However, this evolving mobile Web service space poses a more challenging question. The reasoning is that first, as the diversity in both smart devices and web services with similar functionalities surges, achieving personalization in service discovery will become more challenging due to the complexity of applying context [1], [20]. Second, with smart devices themselves now being able to provide web services to other peers, the anticipated rise in the number of web services will directly translate to more resource burden on the side of the resource-constrained devices. The question, therefore, is: how will service discovery efficiency be guaranteed without necessarily compromising the operational effectiveness of the discovery frame?

Interestingly, the topic of MWS has attracted concerted research attention in the last two decades. Therefore, the challenge stated above only presents an opportunity to build on the existing body of knowledge. While many scholarly service discovery frameworks have been proposed in the literature, the focus has been chiefly on either using lightweight service discovery techniques [1], [20] or leveraging on the flexible and unlimited cloud computing resources to bootstrap Web service discovery [9], [11].

This paper draws insight from these frameworks to extend to the Ad-hoc Mobile Cloud paradigm: "enabling mobile and other smart devices to function as service providers in addition to their conventional role as service requesters" [1]. In extending this concept, this paper

proposes a mobile web service discovery framework where the cloutlet is an operational mediator between mobile providers and clients. With this approach, services can still be made available opportunistically in scenarios where the internet may be inaccessible. Also, the cloutlet's vast storage and processing power are taken advantage of to allow the use of semantic-based service discovery techniques, which otherwise would have posed an unbearable resource burden to resource-poor devices.

2. The service discovery challenge in context

Contextualized service discovery that is, finding web services that best fit users' and devices' contexts, remains a key challenge in mobile environments [11], [21]. This challenge emanates from the characteristic nature of mobile environments – they are highly dynamic and mainly comprised of resource-constrained devices. For instance, because service relevance in such environments is context-dependent, any unanticipated change in context may lead to discovering services unsuitable for the client based on their current context [21]. On the other hand, hosting web services and running the service discovery operation is resource-intensive. Therefore, mobile devices are burdened resource-wise during service discovery because they are battery-dependent and limited in other resources [11], [22]. Research to address these challenges has increased significantly in the last decade, focusing on the concept of personalization or context-awareness, which involves using vital context information to make service discovery processes responsive to the variations in users' settings [23]. Also, to mitigate the resource burden on mobile devices, various ad-hoc mobile service discovery frameworks that essentially employ lightweight techniques and have the architectural capabilities to utilize context information have been proposed in the literature [1], [20]. These scholarly solutions notwithstanding, the evolutionary trends in pervasive computing have further magnified the challenge of realizing the web service paradigm. As highlighted earlier, these trends strongly indicate that mobile devices are increasingly advanced in functionality and capability, and web services supporting technologies are more rampant. And as a direct consequence, it can be anticipated that the above trend will correspondingly bring about an increase in the number of i) more advanced mobile devices, ii) other MWS-enabled devices, iii) diverse web services as well as service providers, iv) and even an increase in the number and diversity of web service clients or consumers.

The scenario highlighted above reflects the emerging context and complexity of the service discovery challenge in MWS environments. The following section contextualizes and discusses this complexity under two key subjects that foreground the framework proposed in this paper:

2.1 Operational effectiveness

To enable mobile devices to provide web services themselves, whether independently or by complementing the cloud, it becomes crucial to have an MWS framework with the design capabilities to support and sustain the critical operations effectively. Fundamentally, these requirements resonate with the central concerns, such as whether the design of the architectural components of the framework aims at compensating for the inherent weaknesses of mobile environments [24]. Notably, such a framework must guarantee that devices do not run out of storage, memory, and battery while discovering or providing a service. And on the other hand, the framework must have the capability to handle scalability and security demands. Ultimately, the challenge here is that mobile devices will always be constrained resource-wise and be battery-dependent. Therefore, they will not always scale optimally as the number of web services grows nor efficiently run energy-intensive semantic service discovery algorithms. As such, hardly would the requirements stated above be practicable. On the one hand, existing solutions have primarily leveraged the cloud system as the backbone mechanism to boost both storage and processing power. However, such an approach, though effective, does not fully realize the goal to ensure uninterrupted availability of web services even when the remote cloud becomes inaccessible due to weak or no internet connection, which is the driving idea behind the MWS paradigm [21].

On the other, using lightweight discovery techniques has been a well-advocated alternative, but which is essentially a trade-off between discovery efficiency and less resource burden. Moreover, uplink connections to the cloud server are often associated with high latency, resulting in enormous energy consumption [17]. Therefore, this paper contributes to extending the existing MWS paradigm by offering an architectural framework of a service discovery mediated by a cloudlet. A cloudlet is a trusted and high capabilities computer connected to the internet [25]. Such a computer also has inbuilt Wi-Fi APs to enable other mobile devices to access it directly. Therefore, by creating a network of peer devices around a cloudlet, these devices can provide, discover, and invoke web services within the ad-hoc mobile cloud without always requiring internet access to connect to the cloud system directly. This approach, aside from mitigating energy consumption, can potentially guarantee scalability by taking advantage of the high capabilities of the cloudlet. Essentially, such an ad hoc MWS provisioning platform can enable peer devices to always run optimally by ceding part of their storage and processing tasks to the cloudlet. This proposed framework builds on and extends an earlier work that implemented and evaluated a context-

aware service discovery model for ad-hoc mobile cloud [1], [22], [26].

2.2 Functional efficiency

As earlier stated, discovering relevant services is one of the crucial challenges in mobile environments in general [11], [22]. However, in the context of MWS, the idea of service relevance is interested not only in discovering services that match users' specific requirements. There is also the aspect of service relevance concerned with the emerging concept of resource-awareness, which measures services relevance by the extent to which the discovered service matches the resource capabilities of devices [27]. Again, the broadening of the scope of MWS potentially extends the complexity of the foretasted challenge. The thinking is that when the diversity of web services and mobile devices increases, extracting and managing context will correspondingly increase. That is, context space is proportional to the diversity of the context dimension.

Therefore, in context, the idea of functional efficiency argues that MWS frameworks must overcome the above complexity as a requirement for discovering relevant web services without overwhelming the resources of mobile devices. Consequently, a twofold solution approach is proposed in this paper to fill in the above gap: ensuring extensive context gathering and employing efficient context processing and application techniques. Based on existing context models, which according to other scholars, are not comprehensive enough, an extended context model is proposed in this paper [1], [7]. Also, using the cloudlet in the proposed framework can support efficient semantic-based algorithms without the mobile devices incurring an unnecessary resource burden

3. The Expanded context model

Since its conception by Schilit [28], the idea of context-awareness has phenomenally influenced the design of smart devices and significantly shaped the model of e-service delivery [29]. Concerning MWS, context-awareness mainly focuses on making the service discovery process more autonomous and responsive to user settings variations. As stated by Hagen [23], context-awareness is "the ability to provide content and services tailored to individuals based on the knowledge about their preferences and behaviour".

Because the core emphasis is about creating systems or services that draw their meaning and essence from the unique world of an individual user (which includes the user), context-awareness is often encapsulated in the term personalization. In this regard, an efficient discovery mechanism guarantees the return of services that best reflect the unique state or situation of the requester. Therefore, current solutions have relied heavily on what Egbe and Murimo [1] described as a "user-centric" approach. That is, driving service personalization based on the user's profile, which comprises user preferences, current activities,

physical location, and surrounding objects. However, such a limited scope of context information fails to adequately capture the emergent context domain, as highlighted earlier. It is in recognition of this weakness that Badidi proposed an extended context model in which service personalization hinges on three components of context: "user's functional needs", "device in use", and the "user's context". The study reported in [1] expanded this model further by adding two more dimensions of context information, namely, device resources and device preferences. In the study, the authors maintained that device resources are a crucial driver of service personalization in MWS environments.

To enrich context information and thereby improve the efficiency of discovering relevant services in MWS environments, a more elaborate context model is presented in this paper, as depicted in Figure 1. The model illustrates how a myriad of high-level context data

can be derived from three primary low-level or parent context information: user-aware, environment, and device-aware context. The idea is that the richer and more comprehensive the context information available and used for service adaptation, the more the discovered services can be tailored to fulfill a particular user's unique state and preferences. In other words, the more relevant the service to the user.

4. Cloudlet-based web service discovery architecture

From the architectural design depicted in Figure 2, the proposed service discovery framework is driven by two independent but complementary clusters of functional components, namely, the Cloudlet components and MWS or ad hoc components.

Figure 1: Comprehensive context model

4.1 The cloudlet components

These are the components that run on the cloudlet. As highlighted in section 2, to ensure the optimal performance of MWS discovery frameworks as MWS proliferate and gathering and managing context becomes more and more complex, such a framework must meet the requirements for operational effectiveness and functional efficiency. Therefore, by design, the approach using the cloudlet to drive part of the framework aims to bootstrap MWS discovery. And as depicted in Figure 2, there are four functional components within the cloudlet:

4.1.1. MWS Database/Directory

These components serve as the MWS repository. Since the proposed framework aims principally at using the cloudlet to alleviate the resource dilemma of mobile

devices, the repository components are employed to host mobile web services directly and indirectly. First, the framework guarantees the scalability of the service provisioning platform by facilitating direct hosting WMS using a web service database cluster [30]. Second, to adapt to the dynamic nature of mobile environments as a strategy to avoid a single-point-of-failure scenario, the cloudlet maintains a directory of MWS hosted on peer mobile devices. Such MWS, though hosted on peer devices, are accessed through the cloudlet's MWS directory using the eXtensible Messaging and Presence Protocol (XMPP), a web service storage model for mobile environments that utilizes [31].

4.1.2. Request listener and router

The cloudlet receives and processes service requests generated and forwarded by various web service client devices via their request manager through this component. Therefore, the request listener functions as a mediator –

takes context enriched service requests and passes them on to the discovery agent.

On the mobile device side, the request listener utilizes relevant technologies to function in a resource-aware mode. For example, the request listener remains inactive except when i) a device is ready to offer a service within the constraints of its resources ii) a client device is forwarding a service request to the cloudlet or returning the list of retrieved services. Moreover, the device offering a service takes advantage of the energy-saving mechanisms supported by the Wi-Fi Direct technology, namely the Opportunistic Power Save protocol (OPS) and the Notice of Absence protocol (NOA) [32].

4.1.3. Discovery Agent

Because of the dynamic nature of mobile environments, the relevance of discovered MWS is highly context-dependent [27]. Therefore, in addition to tailoring services to suit specific clients, most service discovery models have also adopted the context-based service ranking approach to ease service selection and improve user experience. Motivated by that insight, the discovery agent of the proposed framework consist of the service finder and ranker subcomponents [27], [33], [34]. First, when the discovery agent receives a service request, the service finder uncouples the incoming request into its functional and non-function parameters. These parameters are the context-based filters derived from the proposed expanded context model of Figure 1.

Interestingly, it has become possible to support dynamic and proactive web service discovery by leveraging the expressivity and flexibility offered by the WSDL-M lightweight service description standard, which enables diverse context information to be incorporated into web service descriptions [11]. With these parameters, the service finder triggers a service discovery operation on the repositories. The conceptual expression of the process of discovering and retrieving MWS is as given below:

Assuming W_C , W_M , and W_R are the sets of MWS hosted on the cloudlet, peer mobile devices, and on the remote cloud server, respectively, then:

$$W_C = \{W_1, W_2, W_3, \dots, W_n\}$$

$$W_M = \{W_1, W_2, W_3, \dots, W_n\}$$

$$W_R = \{W_1, W_2, W_3, \dots, W_n\}$$

And if W_{RC} , W_{RM} , and W_{RR} represent the sets of relevant

MWS that can be retrieved from the cloudlet, peer mobile devices, and the remote cloud server, respectively, then the set of all returned relevant MWS is expressed as:

$$W_{\bar{R}} = \sum_{n=1}^n (W_{RC}, W_{RM}, W_R)$$

Therefore, after decoupling a service request, the

discovery agent invokes the matchmaking algorithm and applies the extracted service and context information as filter parameters. Then, all matching or relevant MWS are retrieved from the appropriate web service repositories, as illustrated in Figure 2. These MWS are then ranked based on a context-aware similarity index computed by the services ranker [27], [35], [36].

4.1.4. Semantic Reasoner

By interacting with the discovery agent, the semantic reasoner helps to provide a robust and scalable matchmaking algorithm that drives the MWS discovery operations. Although semantic reasoners are resource-intensive, leveraging the cloudlet's vast processing power to use semantic reasoners can be a potential strategy to boost the functional efficiency of the proposed framework. For example, using semantic reasoners significantly enhances precision and recall rate, which are the critical performance metrics of efficient service discovery [37], [38]. More so, the cloudlet's computing power helps shield mobile devices from the associated resource burden.

4.2 The Ad Hoc components

Other components run on various MWS enabled devices, as shown in the proposed framework of Figure 2. Having ad hoc components resonate with the core idea behind the ad hoc mobile cloud paradigm. This paradigm advocates the need for mobile devices themselves to be co-providers of web services in a manner that can create an “opportunistic platform for inexpensive and collaborative resource provisioning” [27]. Such a platform will, amongst other benefits, i) ease the high energy demand of wireless connection from mobile devices to the Cloud, ii) reduce the cost associated with Internet connection charges and capital infrastructure, and iii) most importantly, ensure the availability of web services (with a cluster of mobile devices) whenever the Cloud or cloudlet becomes inaccessible [13]. Except for the request manager and the device monitor, any other ad hoc component is an optimized version but the same function as its corresponding cloudlet component. The optimization is a response to the resource constraints of mobile devices.

4.2.1. Request manager

With this component, clients construct and forward service requests to the discovery engine. The capability of this component employs the JSON string data format, which Wi-Fi Direct supports to build context-rich service requests. Meaning that the request manager serves as an interface for constructing service requests and displaying retrieving returned serves. When the request manager launches a request, the request listener picks up and forwards the request to the cloudlet.

4.2.2. Device monitor

The device monitor is responsible for mining the comprehensive context information offered in the context model of Figure 1 to enhance service discovery efficiency. The device monitor facilitates the extraction of both static and dynamic context through its context and resource manager components. As demonstrated in Figure 1, static context is the category of context that does not change as a function of the mobile environment. Static context includes an aspect of user-aware context (user profile and preferences), service-centred aspect of device-aware context (service profile and identity), and capability-centred aspect of device-aware context (device capabilities). Therefore, they can either be autonomously mined or manually provided and stored in the SQLite database. The proposed context model also offers a second category of context, namely, dynamic context - the aspects of

context information that may change as a function of the intrinsic characteristic of mobile environments. This category of context includes various implicit user profile contexts, environment-aware context (about contexts), and the resource-centred aspect of device-aware context (resource state). Dynamic context constantly changes over time. Therefore, the device monitor extracts it only when processing a service request. The idea is that the relevance of a discovered MWS to its client largely depends on the current state of that device's dynamic context. So, whenever the mobile discovery agent receives a service request, it triggers the device monitor. This action activates the resource monitor component, which helps track the current state of a device's resources (such as available memory, battery level) and other dynamic contexts. The Android utility functions offer vital methods that can be employed to extract these dynamic contexts information.

Figure 2: A Cloudlet-based MWS framework

5. Usage scenario illustration

Following the proposed framework, Figure 3 illustrates the four operational scenarios that can play out

during service discovery.

5.1 Scenario 1: Access cloud service indirectly

Addressing the challenge of occasional inaccessibility of the central Cloud is central in realizing the MWS paradigm. Therefore, it becomes highly imperative for the proposed framework to consider the operational scenario where mobile devices within the ad hoc mobile cloud might not be able to afford the cost of Internet access due to some unavoidable constraints [13]. In the context of this work, the proposed framework adopted the cloudlet approach to bridge the weakness highlighted above. The reasoning is that a cloudlet can be connected to the internet, thereby ensuring that web services are available to other ad hoc nodes in the network. The above idea unveils a scenario where mobile devices that do not temporarily have an Internet connection can still access web services by connecting to the cloudlet via Wi-fi Direct. Moreover, aside from not having internet access, there can be instances where the requested web service may only be found in a remote cloud server.

5.2 Scenario 2: Accessing cloudlet services directly

In addition to having access to unlimited cloud services via reliable internet access, the cloudlet also maintains a complementary repository of MWS, as demonstrated in Figure 2. This repository enables the cloudlet to serve the purpose of an intermediary or buffer MWS provider. This approach aims at avoiding a total reliance on the availability of internet access. On the other hand, it can lighten Internet connection charges and the associated resource burden on mobile devices. So, in the second scenario, a client gets a web service directly from the cloudlet's MWS database when i) the internet access is temporarily down or ii) the requested service is found in the local database.

5.3 Scenario 3: Accessing device-hosted MWS via the cloudlet

One key motivation for this paper is the need to optimize resource usage as a performance enhancement strategy in MWS provisioning and discovery. Consequently, though mobile devices also host web services and can independently perform discovery functions, they by default only initiate service requests and then hand over the actual discovery operation to the cloudlet to save their constrained resources. With this mode of operation, it becomes apparent to envisage a third scenario where the cloudlet functions as a service discovery mediator between two or more mobile devices. In practice, this scenario plays out when the MWS requested by a client referenced in the MWS directory, which is not an actual MWS database but a mapping of

MWS specifications and the respective mobile device(s) providing the service. Therefore, whenever a request matches the service specification(s) on the MWS directory, the cloudlet uses the additional information provided in the directory to link up with the mobile device offering that service to retrieve and return the requested service.

Figure 3: Usage scenarios

5.4. Scenario 4: Access peer-hosted MWS directly

There is also the scenario where peer hosted MWS are accessed directly by mobile clients independent of the cloudlet. Such a scenario is crucial because it is infeasible to assume that the cloudlet will always be functional. So, when the cloudlet is operationally down, mobile devices can take over the discovery operations. The implication is that the cloudlet's downtime does not create a single-point-failure situation where the entire process becomes grounded due to one component's failure.

The usage scenarios of section 5 demonstrate the potential of the proposed framework to enhance effective and efficient service discovery. Overall, aside from having the potential to support a more robust ad hoc service provisioning platform, the operational scenarios highlighted above demonstrate how the proposed architecture's design addresses the critical requirement of eliminating downtime in the service provisioning operation.

For illustration, the implementation of these scenarios will utilize the sequence of functions as shown in Figure 4.

Figure 4: MWS discovery scenario flowchart

6. Conclusion

The idea of empowering mobile devices to collaborate with other peer devices to host and provide web services by forming an opportunistic and inexpensive pool of mobile web services has received increasing research attention in recent decades. Among others, the challenges

of the vast energy requirement of wireless connections and the affordability and accessibility of internet service are the most prominent factors driving the emergent MWS paradigm. Moreover, the proliferation of web services, the advancing capability of mobile devices, and the emergence of supporting technologies suggest that such MWS platforms have the potential to drive the future of e-marketplace in general and SMMEs in specific [39]. Nevertheless, realizing a robust MWS discovery platform is hugely challenged by the dynamics of mobile environments coupled with the inherently constrained resources of mobile devices. These challenges express the need for service discovery models supported by architectural designs grounded on the concepts of resource-awareness and personalization. In this context, this paper makes two significant contributions to the existing scholarship in the field of ad hoc mobile cloud computing in general and the specific domain of MWS discovery and personalization. Specifically, using a graph-based categorization of context, this paper offers an expanded context model. This model allows various high-level contexts information to be derived from three primary low-level or parent contexts. Also, in contributing toward the realization of robust MWS discovery architectures, this study identified, contextualized, and framed the solution to the MWS discovery challenge as the twofold requirement of operational effectiveness and functional efficiency. Above all, this paper offered a conceptual architectural framework that utilizes a cloudlet to support scalable and resource-aware service discovery. By design, this architecture can reduce the traditional resource burden on mobile resources by sharing resource-intensive processes with a cloudlet while also providing alternative MWS repositories to avoid total dependence on the remote Cloud for service invocation. In the future, we plan to extend the implementation of our earlier context-aware discovery model to include the new functionalities proposed here.

References

- [1] D. A. Egbe and B. M. Murimo, "A Resource-aware Service Discovery Architecture for Ad-hoc Mobile Cloud," *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 6, pp. 2402–2408, 2020.
- [2] M. Parameswari and M. B. Moses, "Online measurement of water quality and reporting system using prominent rule controller based on aqua care-IOT," *Des. Autom. Embed. Syst.* 2017 221, vol. 22, no. 1, pp. 25–44, Oct. 2018.
- [3] M. Attaran and J. Woods, "Cloud computing technology: improving small business performance using the Internet," *Journal of Small Business and Entrepreneurship*, vol. 31, no. 6, pp. 495–519, 2019.
- [4] M. Attaran and J. Woods, "Cloud computing technology: improving small business performance using the Internet," *Journal of Small Business and Entrepreneurship*, vol. 31, no. 6, pp. 495–519, 2019.
- [5] M. Choudhary and N. Singh, "Mobile Marketing: Mobile Advertising Strategy," *Journal of the Gujarat Research Society*, vol. 21, no. 14, pp. 337–3457, 2019.
- [6] A. Soni and V. Ranga, "API features individualizing of web services: REST and SOAP," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 9, pp. 664–671, 2019.
- [7] E. Badidi, Y. Atif, Q. Z. Sheng, and M. Maheswaran, "On personalized cloud service provisioning for mobile users using adaptive and context-aware service composition," *Computing*, vol. 101, no. 4, pp. 291–318, 2019.
- [8] S. Al-Janabi and N. Y. Hussein, "The Reality and Future of the Secure Mobile Cloud Computing (SMCC): Survey," in *Lecture Notes in Networks and Systems*, vol. 81, Springer, 2020, pp. 231–261.
- [9] 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*, Dec. 2014, pp. 113–118.
- [10] D. M. Simanjuntak and G. P. Kusuma, "Mobile Application

Using Location-Based Service for Supporting Tourism Industry,” *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 4, pp. 1079–1085, 2020.

[11] K. Elgazzar, H. S. Hassanein, and P. Martin, “DaaS: Cloud-based mobile Web service discovery,” *Pervasive and Mobile Computing*, vol. 13, pp. 67–84, 2014.

[12] C. Bobed, R. Yus, F. Bobillo, and E. Mena, “Semantic reasoning on mobile devices: Do androids dream of efficient reasoners?,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 35, pp. 165–183, 2015.

[13] M. Satyanarayanan, Z. Chen, K. Ha, W. Hu, W. Richter, and P. Pillai, “Cloudlets: At the leading edge of mobile-cloud convergence,” in *Proceedings of the 2014 6th International Conference on Mobile Computing, Applications and Services, MobiCASE 2014*, Jan. 2015, pp. 1–9.

[14] B. Fekade, T. Maksymyuk, and M. Jo, “Clustering hypervisors to minimize failures in mobile cloud computing,” *Wireless Communications and Mobile Computing*, vol. 16, no. 18, pp. 3455–3465, 2016.

[15] Q. Wu and L. He, “Analysis of Social Media User Network Access Behavior Based on Web4.0,” in *6th International Conference on Dependable Systems and Their Applications (DSA)*, Mar. 2020, pp. 127–134.

[16] K. Samdanis and T. Taleb, “The Road beyond 5G: A Vision and Insight of the Key Technologies,” *IEEE Network*, vol. 34, no. 2, pp. 135–141, 2020.

[17] T. L. Tawalbeh, M. A. Tawalbeh, and M. Aldwairi, “Improving the impact of power efficiency in mobile cloud applications using cloudlet model,” *Concurr. Comput. Pract. Exp.*, vol. 32, no. 21, p. e5709, Nov. 2020.

[18] C. Stergiou, K. E. Psannis, B. G. Kim, and B. Gupta, “Secure integration of IoT and Cloud Computing,” *Future Generation Computer Systems*, vol. 78, pp. 964–975, Jan. 2018.

[19] Y. Changsheng, H. Kaibin, and C. Hyukjin, “Energy Efficient Mobile Cloud Computing Powered by Wireless Energy Transfer - *IEEE Journals & Magazine*,” *The Scientific World Journal*, vol. 34, no. 5, pp. 1757–1771, 2016.

[20] N. T. B, V. K, and K. A, “Optimized mobile cloud resource discovery architecture based on dynamic cognitive and intelligent technique,” *Microprocess. Microsyst.*, vol. 81, p. 103716, Mar. 2021.

[21] D. A. Egbe and B. M. Murimo, “Resource-awareness: A Strategy for Resource Optimization and Relevant Service Discovery in Ad-hoc Mobile Cloud,” *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 13, pp. 4029–4043, 2018.

[22] D. A. Egbe and B. M. Mutanga, “Towards a Two-tier Context-based Service Discovery Framework for Mobile Web Services 1,” *International Journal of Pure and Applied Mathematics*, vol. 118, no. 18, pp. 3305–3316, 2018.

[23] B. Henkens, K. Verleye, and B. Larivière, “The smarter, the better?! Customer well-being, engagement, and perceptions in smart service systems,” *Int. J. Res. Mark.*, vol. 38, no. 2, pp. 425–447, Jun. 2021.

[24] C. Stergiou, K. P.-M. Principles, U. Practices, U. And, and U. 2020, “Recent Advances Delivered in Mobile Cloud Computing’s Security and Management Challenges,” in *Modern Principles, Practices, and Algorithms for Cloud Security*, 2017, p. 23.

[25] S. K. Yadav and R. Kumar, “A Mobile Cloud Computing Framework for Execution of Data as a Service using Cloudlet,” *Kuwait J. Sci.*, vol. 48, no. 3, Jun. 2021.

[26] D. A. Egbe, “Context-Aware Discovery of Relevant Web Services in Ad-hoc Mobile Cloud,” *University of Zululand, South Africa*, 2017.

[27] D. A. Egbe, B. M. Murimo, and M. O. Adigun, “A Device Resource-aware Service Discovery Solution for Ad-hoc Mobile Cloud,” *International Journal of u- and e- Service, Science and Technology*, vol. 10, no. 1, pp. 299–316, 2017.

[28] B. Schilit, N. Adams, and R. Want, “Context-Aware Computing Applications, *Mobile Computing Systems and Applications*,” 1994.

[29] A. Vlachostergiou, G. Stratogiannis, G. Caridakis, G. Siolas, and P. Mylonas, “User Adaptive and Context-Aware Smart Home Using

Pervasive and Semantic Technologies,” *Journal of Electrical and Computer Engineering*, 2016.

[30] H. Rojas, K. A. Arias, and R. Renteria, “Service-oriented architecture design for small and medium enterprises with infrastructure and cost optimization,” *Procedia Comput. Sci.*, vol. 179, pp. 488–497, 2021.

[31] R. S. Garba, R. Mohamad, and N. A. Saadon, “Self-adaptive mobile web service discovery framework for Dynamic Mobile Environment,” *J. Syst. Softw.*, vol. 184, p. 111120, Feb. 2022.

[32] G. Han et al., “IEEE Access Special Section Editorial: Emerging Trends of Energy and Spectrum Harvesting Technologies,” *IEEE Access*, vol. 9, pp. 117673–117678, Aug. 2021.

[33] S. Y. Lin, C. H. Lai, C. H. Wu, and C. C. Lo, “A trustworthy QoS-based collaborative filtering approach for web service discovery,” *Journal of Systems and Software*, vol. 93, pp. 217–228, 2014.

[34] W. Serrai, A. Abdelli, L. Mokdad, and Y. Hammal, “Towards an efficient and a more accurate web service selection using MCDM methods,” *Journal of Computational Science*, vol. 22, pp. 253–267, Sep. 2017.

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

[36] S.-Y. Lin, C.-H. Lai, C.-H. Wu, and C.-C. Lo, “A trustworthy QoS-based collaborative filtering approach for web service discovery,” *Journal of Systems and Software*, vol. 93, pp. 217–228, 2014.

[37] A. Zhou, K. Ren, X. Li, W. Zhang, X. Ren, and K. Deng, “Semantic-based discovery method for high-performance computing resources in cyber-physical systems,” *Microprocess. Microsyst.*, vol. 80, p. 103328, Feb. 2021.

[38] M. S. Alshafaey, A. I. Saleh, and M. F. Alrahamawy, “A new cloud-based classification methodology (CBCM) for efficient semantic web service discovery,” *Clust. Comput.* 2021 243, vol. 24, no. 3, pp. 2269–2292, Mar. 2021.

[39] M. Bethel, “Service Discovery in Mobile Ad-Hoc Environments: A Solution Space Analysis,” *academia.edu*, vol. 8, no. 7, pp. 3675–3683, 2020.