

# A Systematic Overview and Comparative Analysis of Service Discovery Protocols and Middlewares

Jawad Hussain Awan<sup>1\*</sup>

Institute of Information and  
Communication Technology  
University of Sindh, Jamshoro, Pakistan

Usman Naseem<sup>2</sup>, Nazar Waheed<sup>4</sup>

University of Technology  
Sydney, Australia

Shah Khalid Khan<sup>3</sup>

RMIT University  
Melbourne, Australia

**Abstract**—Context is major source of communication, where information is gathered easily from user context due to the progress of smart and context aware systems. Even, service directory also supports the systems to response the requests, sent through client. In this paper, the authors overviews context aware systems, their sensing capabilities in location or beyond location along with COIVA (a context aware system). Eight discovery protocols along with their functionalities (such as DEAPspace, DNS-SD, JXTA, RDP, LDAP, CORBA Trader, UDDI and Superstring) are discussed and compared to evaluate the performance and efficiency of system. In addition, six middleware (such as CAMPUS, CASE, SeCoMan, CoCaMAAL, BDCaM, and FlexRFID) are compared to evaluate factors (such as Architectural style, Context abstraction/Reasoning level, Context awareness level, Contextual adaptation approaches, Decision making, and Programming model). The authors further categorized them into sub categories discussed in Section 4 and named CoCaMAAL as better middleware as compared to others.

**Keywords**—Pervasive computing; service discovery protocols; context-aware; middleware; privacy

## I. INTRODUCTION

In Modern advancement in Pervasive computing brings a number of challenges like heterogeneity, scalability, privacy and more. It is possible because of code, user and device mobility. Hence, the research opportunities may support to discover new applications in environment with the help of these factors. The interaction between resource and user takes place in ad-hoc, permanent and transience nature of computing environment [1]. In this way, a number of applications are proposed having concept of location and context awareness. Context awareness not only enables the context aware systems, but also progresses the applications, such as context-aware recommendation, context-aware reminders, and intelligent call redirection [2], [3]. Human imagination persistent the invention and aggressive environment creates delay in pervasive computing. Hence, there is no any technical solution for human interface and complexity [4]. Thus, the main objective of pervasive computing research is to find and discover solution for above highlighted issues as invention may progress. Thus, discovery protocols are proposed to discover, advertise and register resources, services for clients or users. The main theme of this paper is to provide an overview and comparative analysis of service discovery protocols and middlewares. This paper also comprises of five sections. Section II discusses and compares the service discovery protocols. Section III discusses context-

aware and its existing systems, while Section IV compares and discusses middlewares, while section V provides discussion regarding critical analysis of service discovery protocols and middlewares. Section VI concludes the research precisely.

## II. SERVICE DISCOVERY PROTOCOLS

In this section, following service protocols are discussed and illustrated.

### A. DEAPspace

The DEAPspace protocol is developed by Institute of Business Management (IBM) to focus small similar scale networks of Bluetooth. Specially, DEAPspace priorities to all nodes as being available in broadcast range while Konark rely on the TCP/IP rather than IP. DEAPspace also utilize a dynamic slotting of broadcast scheme, and pro-actively pushes the advertisements within the network. Hence, the resources knowledge is available among nodes of network.

DEAPspace protocol uses input or output schemes to define the services. The description of scheme is dependent of MIME and hierarchical in nature [5]. Though, each element may comprise of attributes. Like “Application → PostScript →version2”, the PostScript element has attributes like color = value and ppm = value. The value for color may be Yes or No while ppm (Pages Per Minute) value may be a numeric value. For example, color=no and ppm=10. Furthermore, DEAPspace neither supports query relaxation nor expressions over the attributes.

### B. DNS-SD (Domain Name System - Service Discovery)

DNS-SD is an Apple’s Rendezvous technology and offers service discovery functionality for [6]. At initial level, these devices uses ZeroConf draft IETF standard to assign IP addresses from link-local range. Whenever, a node lacks IP addresses and creates conflict with peer nodes in network. Hence, each node assigned an IP address to avoid conflict among peers as well as uses DNS-SD in service allocation. Each node or device hosts a DNS server and clients to register or locate services on the network as easily discovered by other clients whenever requested utilize multicast (mDNS-SD) messages. Though, Apple provides well-known implementation of DNS-SD in Rendezvous technology, and DNS scheme is used to construct DNS-SD. New message is not defined in DNS scheme while scaled to internet. The scalability feature is gained at particular domain while DNS-SD does not provide service discovery to use cases, relax

\*Corresponding Author

queries. Therefore, it relies on DNS defined messages. It is also notified from the research that DNS-SD is only suitable for IP based environments and requires suitability of other than IP supported environment.

### C. JXTA (Juxtapose)

JXTA is Peer-to-Peer (P2P) technology and build by Sun Microsystems [7]. This technology is helpful to design P2P applications. Further, JXTA is composed of following protocols and their features illustrated in table no such as Peer Discovery Protocol (PDP), Custom Service Discovery Protocol (CSDP), Higher-level Service Discovery Protocol (HSDP), Peer Resolver Protocol (PRP), and Rendezvous Protocol (RP).

PDP is responsible to discover resource for an advertisement within peer group. The resources include peers, peer-groups, pipes and modules [8]. CSDP is implemented within non-world peer-groups, and then PDP works for bootstrapping purpose. If custom discovery protocol is unavailable then PDP is responsible for discovering of resources [9]. HSDP is utilized to process detailed discovery information. The information includes discovery of queries and advertisement and both are based on Extensible Markup Language (XML). In the definition of HSDP, reduction in the interaction and increase in scalability ratio may take place in peer groups. In few applications, PDP is allowed to discover resources as well [10]. PRP is responsible to route query and its response in peer group of JXTA. The PRP also forwards every method to a specified handler, that handler has semantic definition of message, the message is then sent to peer or a group peers within network [11]. The sending and receiving of message is the function of RP. RP also locates resources and limits the scalability also. Few nodes become rendezvous peers and propagate the messages to subscribed peers by them in network or peer group [12].

Table I compares five JXTA protocols such as PDP, CSDP, HSDP, PRP, and RP having Peer Group, Non-Peer Group, Detailed discovery information, Rendezvous Peers, Semantic definition, Scalability, Resource advertisement, and Resource discovery. PDP has four maximum features capability such as Peer Group, Non-Peer Group, Resource

advertisement, and Resource discovery as compared to other protocols. Hence, PDP is favorable than others.

### D. RDP (Remote Desktop Protocol)

RDP protocol is designed by Perkins and Harjono [13], used for fixed network mobile nodes. These nodes move from one network to another network like when an employee moves his/her laptop or handheld device from office (LAN network) to home (LAN network) or vice versa . The RDP uses a bootstrap approach to centralize a database, resides in the network. In this protocol, resource description uses URL and keywords as forms and queries are based on URNs (Unique Reference Numbers). URN consists of service information, optional resolution path, optional naming authority, and keywords. The initial section of URN describes the service information such as type. URN may be  $n^{21}$  or  $n^{2c}$ .  $n^{21}$  represents that one matched resource must be returned while  $n^{2c}$  represents all matched resources. DHCP provides the address of resource database and it is overridden by optional resolution path. Naming authority name the institution where mobile device discovers itself. Naming authority is also interpreting mechanism of scheme field. The designed protocol uses UDP (User Datagram Protocol) to deliver registration description and advertisements for queries. This protocol is also applicable for IP based environment like DNS-SD.

### E. LDAP (Lightweight Directory Access Protocol)

LDAP is an easy version of the X.500 standard [14]. LDAP is a programming model to design discovery or registry service while it lacks the features of discovery protocols. LDAP is suitable for distributed environment and provides important features utilized in resource discovery. Such as duplication, exchange, and security. Though LDAP is mainly a directory consists of data types, objects (limited to a directory entry). LDAP is also utilized as a resource directory and discovery protocol. In this way, services are registered in directory and clients search or discover them for request of resource [15], [16]. For example, Java objects are stored or registered in the LDAP directory for a retrieval request. The schema is only source to describe that how objects can be stored in directory.

TABLE I. JXTA PROTOCOLS AND FUNCTIONALITY

Ref	Protocol	Peer Group	Non-Peer Group	Detailed discovery information	Rendezvous Peers	Semantic definition	Scalability	Resource advertisement	Resource discovery
[8]	<b>PDP</b>	√	√	×	×	×	×	√	√
[9]	<b>CSDP</b>	×	√	×	×	×	×	√	√
[10]	<b>HSDP</b>	√	×	√	×	×	√	×	×
[11]	<b>PRP</b>	×	×	×	×	√	×	×	×
[12]	<b>RP</b>	√	×	×	√	×	√	×	×

### F. CORBA Trader

Offering and discovering a service is primary function of discovery protocols. Various institutes developed their own protocols for these services like Open Distributed Processing (ODP) trading function [17]. ODP has both features such as offering and discovering a service. Though, these services are known as exporting and importing capabilities. Generally, ODP is a service model but not implemented yet while CORBA trading function is implemented for importing and exporting capability. Like other protocols, CORBA trader is based on five components. Lookup, Link, Register, Proxy, Admin. The clients uses Lookup component to find services. These services are advertised or registered with the help of Register component in the Trader. Then, Link component is responsible to perform internal operation of Traders. While, admin component is also responsible to set trader policies and the legacy services are wrapped or hidden by Proxy component. CORBA trader is suitable for only static networks where nodes are known or defined while unsuitable for dynamic networks like MANETs (Mobile Ad hoc Networks). In this discovery model, services are chosen via granularity rather than query relaxation.

### G. Universal Description, Discovery and Integration (UDDI)

UDDI is a general arrangement in which businesses are enabled to find offered services along with businesses. In this arrangement, when a new or suitable service is offered or found, then, that service is integrated into application. UDDI has four types of entities (Business entity, Business service, tModel), describe the UDDI information and presented in XML. Business entity is at upper level and contains the information of name, address, service of business. At least one or more services are registered in each business entity and combines similar web services offered by businesses [18]. Information in UDDI is described by four entity types, each of which is represented in XML. The business Entity is the top-level structure. It contains information such as the name of the business (or other entity, such as a department within an institution), its address and the type of service the business

provides. Each business Entity contains one or more business Services. This entity type logically groups a set of related web services provided by the business. The business Service is a descriptive structure. It lacks technical information but bind with template structures, which contains technical information like how to invoke or interact or respond the web service. At last, tModel is existing structure outer side of the hierarchy. The structure of tModel utilizes and defines reusable components.

### H. Superstring

Superstring is most efficient and appropriate for static and dynamic environments because it defines three components (two routing protocols and one API for resource or service discovery) in environments [19]. Superstring also scales the number of nodes or devices from computers to handheld or portable devices. The dynamic network deploys a routing protocol that progresses the network to decrease the query processing time for services or resources and adjusts to change in the network. While static network also deploys a routing protocol, that is responsible to scale the resources in wide area network or environment. The deployed protocol consists of various numbers of resources. Least dynamic nature increases the scalability ration in network.

The description of Superstring is efficient and hierarchical. Thus, the description contains a description model (This model has simple and easy expression language, query relaxation and reserved elements). A set of primitives are also defined by Superstring in the description language, responsible to allow queries and advertisements. Hence, queries and advertisements are issued to rapidly convey context-awareness to the applications existing in environment. Table II compares eight service discovery protocols such as DEAPspace, DNS-SD, JXTA, RDP, LDAP, CORBA Trader, UDDI, and Superstring. The authors identified that superstring is better service protocol fulfills the functionality and features, i.e. IP based Environment, Other than IP based Environment, Fixed network, Distributed Environment, Dynamic Networks, Scalability, Advertisement, and Query relaxation.

TABLE II. SERVICE DIRECTORY PROTOCOLS AND FUNCTIONALITY

Ref	Protocol	IP based Environment	Other than IP based Environment	Fixed network	Distributed Environment	Dynamic Networks	Scalability	Advertisement	Query relaxation
[5]	DEAPspace	×	×	×	×	×	×	×	×
[6]	DNS-SD	√	×	×	×	×	×	×	×
[7]	JXTA	×	×	×	×	×	×	×	×
[13]	RDP	√	×	√		×	×	×	×
[14]	LDAP	×	×	×	√	×	×	×	×
[17]	CORBA Trader	×	×	×	×	×	×	×	×
[18]	UDDI	×	×	×	×	×	×	×	×
[19]	Superstring	×	×	√	-	√	√	√	√

### III. CONTEXT AWARE SYSTEMS

In the research, context may be defined in different ways as per situation or circumstances. The initial definition of context was defined by Schmidt [20] as “A context describes a situation and the environment a device or user is in. A context is identified by a unique name. For each context, a set of features is relevant. For each relevant feature a range of values is determined by the context”.

A researcher Dey [21] also defined context in semantic research that “Any information that can be used to characterize the current situation of an entity”. Two other researchers Schilit and Theimer categorized the context into four categories such as Computing, user, physical and time context. They also defined as “Computing context: network connectivity, communication costs, communication bandwidth, nearby resources such as printers, displays, and workstations. User context: the user’s profile, location, people nearby, the current social situation. Physical context: lighting, noise levels, traffic conditions, and temperature [22]. Time context: time of a day, week, month, and year” [23] [24].

Later on, Schilit, Adams and Want [25] defined context aware system as: “A system is context-aware if it uses context to provide relevant information or services to the user, where relevancy depends on the user's task”.

Context aware system or context awareness is backbone of pervasive computing. These systems consists of various functions and some of them are as under [26] and shown in Fig. 1.

Sensing the Context such as location sensing (Indoor and Outdoor). Sensing Low-level Contexts beyond Location such as Time context, nearby objects, Network bandwidth, Orientation, and other low-level contexts. Sensing High-level Contexts such as user’s current activity. Sensing Context Changes such as moving of a person from one location to another.

#### A. Sensing the Context

The basic function of the context aware system is to sense the context. The context may be location, time, user and computing context. Thus, this mechanism is mostly available in these systems to sense and then deliver to application as execute the task as per flow or function [27]. The location is important context to know the user movement from one location to other location. This context varies as user moves and it is easy nowadays to collect or gather user location information because he/she allows the devices to supply their location to applications [1]. User cooperation supports the application or system to be accurate and reliable. If the location sensing uses automatic technique then the system is independent of user and sense the context by applied mechanisms. Such as, an employee is entering in his/her office and press the fingerprint for authentication. His/her location is collected automatically from the system after pressing the fingerprint. The location can be sensed in two modes either indoor or outdoor.

GPS (General positioning System) is best choice for outdoor positioning [28]. The Government of US allowed the

GPS signal at 10 to 20 meter range to achieve more accuracy then earlier [29]. Various applications such as automobile navigation systems in computing environment get benefit from the new policy because tiny and inexpensive devices lack the capability of GPS. Bulusu and researchers proposes a connectivity based localization technique [30]. In this proposed technique, the accuracy may achieve within the range of 3 meters (if the reference points are known). However, GPS is not appropriate in indoor applications because the GPS signals have low signal strength in indoor physical space [31]. Thus, the signals penetrated in the buildings and make unreliable and fluctuating readings due to multipath reflections. Moreover, it becomes a challenge to build an ideal indoor service that is inexpensive, scalable and vigorous with lofty update rate of spatial information [32]. That’s the reason that most of indoor research projects have their own location tracking systems such as Active Badge, Cyber guide, Shopping Assistant and 3D-iD. Few of them uses IR (Infra Red) and others RF (Radio Frequency) [33] [34].

#### B. Sensing Low-Level Contexts beyond Location

Location is not only the context but time, nearby objects, network bandwidth, and orientation are also low-level contexts. The contextual information of time is not difficult to achieve. Most of the systems have capability of built-in clocks, few uses timestamp like Active badge. Time-of-day, day, week, month, year, season, time zone, and onward are different forms of contextual information [35]. Nearby objects are also sensed by the contextual systems. It is also easy to find nearby objects via projection from the existing dataset of systems’ database because the computing environment records the location of people and objects which became part of that system. Such as Teleporting system and the context-aware Pager are good examples of sensing nearby objects.

Besides location, time and nearby objects, network bandwidth is also a chief component of computing and a significant computational context too[36]. The change in bandwidth is not easy method without the support of system. Hence, system support is necessary to get acknowledge when change in network bandwidth takes place. Few systems at user level like Odyssey system [37] uses API calls and others at kernel level like Congestion Manager uses up calls[38] to measure the bandwidth and notify the changes when occurs.

In addition, few low-level contexts can be embedded with system to measure or know physical contexts like intensity of light, vibration, sound, temperature in indoor environment. The researchers or project team may deploy bi-sensor or multi-sensor prototypes to sense more than one context at a time. TEA project has also deployed multi-sensor prototype in their research to measure multi contexts [39]. The research also notified [40] that user’s cooperation is necessary while enhancing the sensing of contextual level in mobile devices because the addition of sensors reduces the user’s mobility due to supplementary size and weight. Further, the research is going on to reduce the deployment of sensors within mobile devices in the computing environment as per user needs or specification [26].

### C. Sensing High-Level Contexts

Like low level context, high level context sensing is also function of context aware systems. Sensing high level context is an emerging challenge for researchers. In this connection, three approaches have been proposed [41] to find user activity. Primary approach is to machine vision while secondary approach is to consult user's calendar and tertiary approach is using Artificial Intelligence (AI) [42] [43]. Primary approach is dependent on computer vision and image processing technology to sense complex social context. Secondary approach uses user's calendar that supports the systems to measure user activities. Tertiary approach with the help of AI recognizes the context by gathering multiple low level context sensors to get contextual information [44].

### D. Sensing Context Changes

The context aware systems not only sense the low level and high level contexts but also sense the changes occurred in system. Various context-aware applications have feature to notify the contextual changes. Generally, a source monitor is defined to poll the present contextual information and then shares the changes occurred in context services. The services have the ability to publish-register-discover or notify interfaces. Once, the changes take place in context and then the information is discovered or notified by registered client or user from context service. Every context has its own properties the location of an employee changes in office building as per necessity while the location of scanner remains same. Thus, each context has different polling rates [45], [46].

Sometimes, the contextual information of context-aware system does not satisfy the clients or users because the system is tracking the location tracking device worn to client or user in indoor environment but that is kept on table or other place for a short period and forgot to wear. In such cases, it is inappropriate for applications.

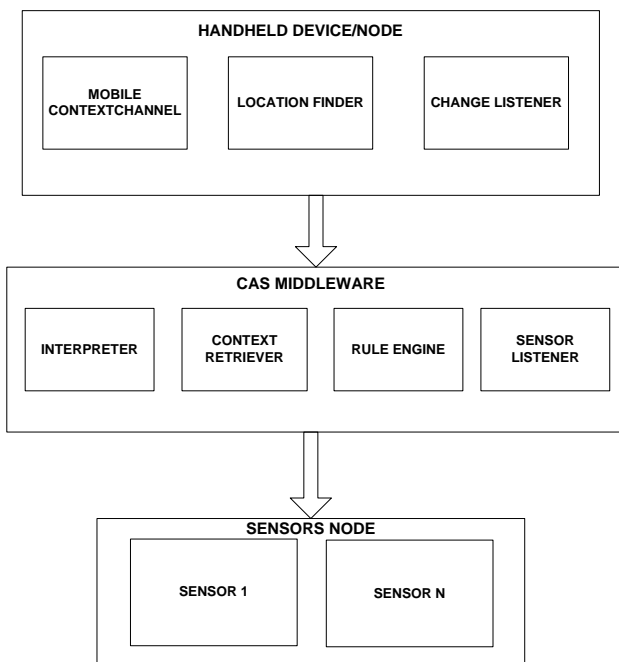


Fig. 1. Architecture of the Context-Aware System [34].

### E. Context-aware and Ontology-Powered Information Visualization Architecture (COIVA)

COIVA is a context aware system that provides the infrastructure to initiate context-powered services via computing environment and smart devices. The semantic and information of contextual elements is shared in system to measure user needs and then adaptive services are offered where required. The COIVA architecture is comprised of two blocks such as COIVA core, and functional engine [47].

1) The COIVA core is a chief component of this architecture. This component is based on context model, which comprises of general and specific sub-models. General model is further categorized into four sub-components such as user, services, environments and devices. Ontology describes each sub-component to discover main elements of distinctive user-centered intelligent environments.

2) Functional engine is the second component of COIVA. It comprised of three modules. Initial model is used to collect and unify context from intelligent systems. This model also plays a vital role in implementation and covert gathered information into contextual information. Intermediate model works as expert system to assume context and evade variations. The last module grips the meta-information (A piece of information that describes a contextual feature and its associations with additional information) [47].

## IV. MIDDLE WARES

Middlewares play an important role as middle man to control, monitor, supervise and maintain the systems or application smarter in context aware environment. Following are few modern Middleware system services [48].

### A. CAMPUS

Context-Aware Middleware for Pervasive and Ubiquitous Service (CAMPUS) [49] is an important middleware and used at runtime for automated context-aware adaptation decision making. CAMPUS is dependent of three main technical approaches such as ontology, descriptive reasoning and compositional adaptation. Generally, the middle-wares depend on pre-defined decisions or policies in dynamic environment while CAMPUS has diverse feature to make decisions dynamically and varies with contextual changes [50][51]. The CAMPUS has three tiered architecture, comprises of programming layer, knowledge layer, and decision layer as shown in Fig. 2.

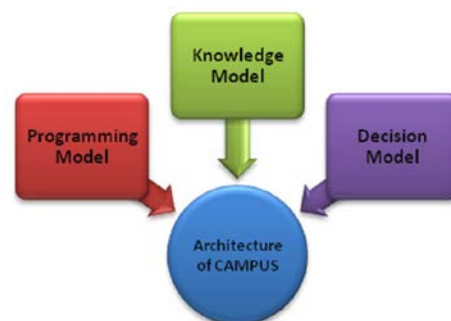


Fig. 2. Three Tiered Architecture of CAMPUS.

The decision layer is basic and an essential layer of CAMPUS. This layer deploys a multi-stage decision model. This model includes screening, selection and preprocessing, to prefer the finest tasklet substitute for a specified task. The automated decisions are taken at this layer, and then forwarded to the second layer named programming layer [52]. The programming layer receives adoptive decisions from decision layer and then reconfigures or constructs context-aware applications by adopting the forwarded instructions from decision layer[53]. The knowledge layer is responsible to represent knowledge semantics and comprises of three models or ontologies such as Service, Context and Tasklet. These ontologies are necessary for CAMPUS in making adoptive decisions. The represented semantics of knowledge are also utmost requirements, may include the required properties of context or existing or run-time context [54].

The CAMPUS initially developed in JAVA SE 1.6 with 1.5.1 Pallet (descriptive reasoning), and JESS 7.1 P2 (Logical Reasoning) [55]. This middleware provides dynamic adoptive decisions and integrate them with context-aware applications. Even though, the security or privacy is emerging issue of CAMPUS and other middleware or context awareness systems [56][57]. The future extension of CAMPUS may adopt security and advanced dynamic adaptation techniques or approaches for context-aware systems.

### B. Context-Aware Services Framework (CASF)

CASF is a middleware and it is presented to provide a variety of context aware services. In addition, the architecture of CASF includes a service directory along with capabilities of composition as well. Because, it was noted that a number of context aware systems lacks these both capabilities. Thus, proposed and presented in CASF. This framework also supports automatic service discovery and integration. Hence, it is based on semantic services to achieve service integration, selection and gathering contextual information while CASF separates the services and contextual information of systems.

The main core of CASF architecture plays an essential role. Thus, it is called as Context Mediation Framework (CMF). It also includes three layers Such as Physical sensor, Public context and contextual service as illustrated in Fig. 3. Physical sensor layer is the basic contextual source of CASF, due to recognition nature. It recognizes the sensor data and categories as per service or system requirement. Public context layer includes two sub layers for contextual information and their complexity. First basic sub layer processes and provides only sensor data while the second one provides both sensor and contextual information together to other contextual information providers. Duse to semantic nature, all contextual information is gathered, processed and generated in this layer. With the help of ontology and OWL-S, the web services are constructed. Hence, interoperability and openness are achieved. The last layer of CMF is Context service layer. This layer consumes the contextual information. As, new context aware services are to be produced to fulfill user requests.

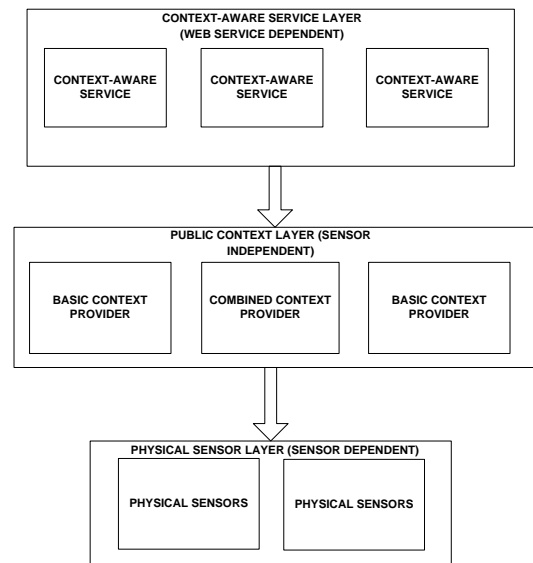


Fig. 3. Features of CASF.

The main uniqueness of CASF is the implementation of the idea of semantic web services. The contextual information is published on the basis of semantic web services. Besides that, this approach becomes reasonable to accomplish automatic discovery and assimilation for contextual data or information. In addition, advanced level protocols and ontologies have to be studied as contextual information is translated into web services. Like SOAP, communication protocols are to be specified or chosen to make communication as effective in between physical sensor and public context layer. However, CASF architecture also lacks the prototype of real environments for context aware services or systems.

### C. Semantic Web-based Context Management (SeCoMan)

SeCoMan is proposed to offer a privacy solution in the development of context aware smart applications or services. While ontology is chief component in the development of SeCoMan as the description of entities be modeled as per requirement like obtaining functional knowledge, and specify the policies of context aware. The architecture of SeCoMan is categorized into three layers, including Context Management layer, Application layer, and Plug-in layer. Application layer is first and top level layer in which various applications reside to offer requested services of users. Context Management layer is the heart of the SeCoMan as well as allows the support to contextual applications. In addition, SeCoMan is divided into three different actors along with specific rights. Users, Application supervisor and framework supervisor are defined actors. However, applications having predefined queries are allowed to receive contextual information either indoor or outdoor as well as semantic ontology is also employed to define policies regarding privacy, location and authorization access.

Plug-in layer is third and last layer of SeCoMan framework. This layer is particularly focused on contextual locations and offers contextual information of SeCoMan framework. Plug in works as an independent source of contextual information. Generally, SeCoMan has limited solution to provide only contextual location. Hence, privacy is

provided to users. Thus, they easily share their location to receive context aware services. The future of SeCoMan is under research to emerge cloud and distributed computing with SeCoMan architecture. This feature will enhance the capabilities of context aware systems.

#### D. CoCaMAAL (Cloud oriented Context aware Middleware in Ambient Assisted Living)

Forkan and other colleagues [58] presented and proposed the CoCaMAAL. The main objective of this research is to enhance the capability of biomedical sensors, because they lack processing power. This feature is necessary for AAL to achieve data aggregation and key monitoring. Besides this, cloud and distributed computing are emerged to perform computational tasks or needs. This middleware became an ideal, easy in data gathering and processing. Service Oriented Architecture (SOA) is soul of COCaMAAL because all the functions (such as modeling, adaptation, mapping, service distribution of context and more) are performed. The proposed middleware is hardware based architecture, also comprised of Body Sensor Network (BSN), supervision systems to fulfill user needs or requirements. Moreover, CoCaMAAL comprised of Context Aggregator Provider (CAP), Service Provider (SP), Context aware Middleware (CaM), and Data Visualization Approach (DVA). CAP is an intermediate tool that converts and abstracts high level of contextual information for AAL. In addition, contextual data is categorized either in pre-defined ontology or sensor based data. Besides, various contexts are emerged to execute and complete the required information. CAP is responsible for the execution of whole process. SPs are the general applications, which produce, generate and manipulate the services for user needs. CaM is used to identify assistive services for collected context and associated actions. CaM is also an essential tool of CoCaMAAL due to execution of key functions such as mapping, management, storage, and retrieval. DVA also plays an important role in utilization of user interfaces. The author developed the architectural prototype in Java language. The implementation and experiments have been done to measure response time, influence level while increase in context takes place. The results illustrated that CoCaMAAL is efficient and effective for AAL environment. The adaptation of computing technologies is also novelty in this research or prototype. However, Reliability, conflict among contextual systems and privacy are also under research and needs improvements in highlighted issues.

#### E. Big Data for Context Aware Monitoring (BDCaM)

BDCaM is an extended version of CoCaMAAL and proposed by [59]. BDCaM is advanced middleware and used as supervising tool for Context-aware systems. This middleware demonstrates a supplementary feature such as personalized knowledge discovery as compared to CoCaMAAL. In this feature, the knowledge is learnt from collected data, which are anomalies of specific patient. The adoption technique and methodology are different than COCaMAAL [60]. Both are essential in decision making while contextual data is collected from context-aware systems. Correlations and Supervised learning are two approaches used to perform functions of BDCaM. Initially, Correlation takes

place between the attributes of contextual information and values of threshold. Map Reduce Apriori Algorithm (MRAA) is applied to generate associations of patient-tailored [61]. The generated rules are used by supervised learning to manipulate collected contextual data.

The working architecture of BDCaM is also split into different distributed or cloud based components. Like AAL, CA, CP, CMS, SP and more. A prototype was also developed for health monitoring systems to measure the functionality of middleware. The results proved that the middleware has detection efficiency among anomalies. Security and privacy is also a challenge for context-aware systems [62]. The emerging of this middleware to other domains is also under research and not yet been suited or implemented.

#### F. FlexRFID

FlexRFID is modern and advanced middleware from discussed above architectures [63]. The aim of this architecture is to offer a policy-based solution in the development and implementation of context aware systems or applications and emerging diverse nodes. FlexRFID is multi-layered middleware, adopts Ponder as PSL (Policy Specification Language) as well as consists of Device Abstraction Layer (DAL), Business Event and Data Processing Layer (BEDPL), Business Rule Layer (BRL), and Application Abstraction Layer (AAL) layers. DAL abstracts the interactive activities among the devices, nodes, and communication medium. BEDPL offers Contextual Information Management (CIS) (Such as aggregation, revolution and broadcasting). BRL copes with policy based operations), and at last, AAL allows communications amid applications and the FlexRFID.

According to literature, it is claimed [64] that FlexRFID offers filtration, grouping, integrity, removal of duplications. Hence, it can be said that FlexRFID is an enabled solution for synchronized communication among emerged technologies of middleware. In addition, Policy enforcement is also feature that differs FlexRFID with other middleware. Various policies such as abstract policy, system policy, ensure security, access control, and other customized services are defined for architecture. The authors have taken two scenarios for experimental work during their research, the results illustrated that response time and volume of policies are directly proportional as well as dependent to each other. If one increases and second one results an increase also. Recent version of FlexRFID only provides necessary privacy mechanisms to specify policies of access control. Application authentication, privacy at sensor nodes and tags, integration of FlexRFID with other distributed or cloud services are to be improved at advanced level for specific applications.

Table III compares six middlewares such as CAMPUS, CASF, SeCoMan, CoCaMAAL, BDCaM, and FlexRFID having Architectural style, Context abstraction/Reasoning, Context awareness level, Contextual adaptation approaches, decision making, and Programming model. CAMPUS is found to be an effective and dynamic middleware, which is suitable for Semantic, Sensing, Parameter adaptation, and Contextual Reconfiguration.

TABLE III. COMPARISON OF MIDDLEWARES

Ref	Middleware	Architectural style		Context abstraction/Reasoning				Context awareness level		Contextual adaptation approaches			Decision making	Programming model
		Layered	Distributed	Key value	Ontology	Markup	Rule	Medium	Semantic	Sensing	Parameter adaptation	Contextual Reconfiguration		
[65], [66]	CAMPUS	√	×	×	√	×	×	×	√	√	√	√	Dynamic	Semantic-based DL
[57] [67]	CASF	√	×	×	√	×	×	×	√	√	√	×		ontology and OWL-S
[68]	SeCoMan	√	×	×	√	×	×	×	√	√	√	×		Ontology / Semantic
[58]	CoCaMAAL	×	√	×	√	×	√	×	√	√	√	√	Dynamic	
[59], [69]	BDCaM	×	√	×	√	×	×	×	×	√	√	√	Dynamic	
[70], [71]	FlexRFID	√	√	×	×	√	√	√	×	√	×	×	Static	Policy based

### V. DISCUSSIONS

The discussions section provides the comparative analysis of three major features and services of pervasive computing field. Table I compares five JXTA protocols such as PDP, CSDP, HSDP, PRP, and RP having Peer Group, Non-Peer Group, Detailed discovery information, Rendezvous Peers, Semantic definition, Scalability, Resource advertisement, and Resource discovery. PDP has four maximum features capability such as Peer Group, Non-Peer Group, Resource advertisement, and Resource discovery as compared to other protocols. Hence, PDP is favorable than others. Table II compares eight service discovery protocols such as DEAPspace, DNS-SD, JXTA, RDP, LDAP, CORBA Trader, UDDI, and Superstring. The authors identified that superstring is better service protocol fulfills the functionality and features, i.e. IP based Environment, Other than IP based Environment, Fixed network, Distributed Environment, DynamicNetworks, Scalability, Advertisement, and Query relaxation. Table III compares six middlewares such as CAMPUS, CASF, SeCoMan, CoCaMAAL, BDCaM, and FlexRFID having Architectural style, Context abstraction/Reasoning, Context awareness level, Contextual adaptation approaches, decision making, and Programming model. CAMPUS is found to be an effective and dynamic middleware, which is suitable for Semantic, Sensing, Parameter adaptation, and Contextual Reconfiguration.

### VI. CONCLUSION

Latest development in smart systems, context aware systems and middleware made real time environments smarter and efficient for users. Emergences of sensors technology have grown user interaction with systems and technology. Likewise, new policies, rules and procedure are being defined to enhance the capabilities and features of smart system. Hence, physical objects are moved to smart objects. In addition, real time data and intelligent data combined to achieve more accuracy and efficiency in such systems, connected components or nodes can be identified via

embedded systems. These systems may communicate to each other via distributed systems or infrastructure. According to Cisco IBSG, the IoT world will include more than 50 billion objects in 2020. In this paper, we presented service discovery protocols and functionality is compared in Section 2, Section 3 overviews context aware systems, their features and discusses few context-aware systems. In addition, middlewares are also discussed and compared to identify most suitable one in Section 4. And discussion is presented in Section 5.

### REFERENCES

- [1] J. H. Awan, S. A. Memon, N. A. Memon, R. Shah, Z. Bhutto, and R. A. Khan, "Conceptual Model for WWBAN (Wearable Wireless Body Area Network)," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 1, pp. 377–381, 2017.
- [2] U. Naseem, S. K. Khan, M. Farasat, and F. Ali, "Abusive Language Detection: A Comprehensive Review," *Indian J. Sci. Technol.*, vol. 12, no. 45, pp. 1–13, 2019.
- [3] U. Naseem, "Hybrid Words Representation for the classification of low quality text," 2020.
- [4] M. Rosic, S. Mladenovic, and L. Borojevic, "Information system user interface complexity," in *Symposium of the Austrian HCI and Usability Engineering Group*, 2010, pp. 509–512.
- [5] J. D. Cormie, A. K. Fischman, and A. H. Vermeulen, "Dynamic application instance discovery and state management within a distributed system." Google Patents, 2018.
- [6] R. Droms and T. P. Donahue, "Visibility control for domain name system service discovery." Google Patents, 2018.
- [7] M. Amoretti and F. Zanichelli, "P2P-PL: A pattern language to design efficient and robust peer-to-peer systems," *Peer-to-Peer Netw. Appl.*, vol. 11, no. 3, pp. 518–547, 2018.
- [8] S. Mallik, R. Palanki, D. P. Malladi, and N. Bhushan, "Hybrid modes for peer discovery." Google Patents, 2019.
- [9] Y. Amishav, E. J. Barkie, O. Dubovsky, and B. L. Fletcher, "Location-based domain name system service discovery." Google Patents, 2018.
- [10] S. Abdellatif, O. Tibermacine, and A. Bachir, "Service Discovery in the Internet of Things: A Survey," in *International Symposium on Modelling and Implementation of Complex Systems*, 2018, pp. 60–74.
- [11] F. Battaglia and L. Lo Bello, "A novel JXTA-based architecture for implementing heterogenous Networks of Things," *Comput. Commun.*, vol. 116, pp. 35–62, 2018.



- [12] S. Kadam, D. Prabhu, N. Rathi, P. Chaki, and G. S. Kasbekar, "Exploiting group structure in MAC protocol design for multichannel Ad Hoc cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 893–907, 2018.
- [13] A. T. Fausak, O. Rombakh, C. D. Robison, C. A. Andrews, and others, "System and method for remote access to a personal computer as a service using a remote desktop protocol and windows hello support." Google Patents, 2019.
- [14] M. A. Thakur, S. G. Bari, R. Deshmukh, and S. Auty, "A Survey of Directory and Database Protocols for Data Extraction," in *Information and Communication Technology for Sustainable Development*, Springer, 2020, pp. 315–325.
- [15] S. K. Khan, M. Farasat, U. Naseem, and F. Ali, "Performance Evaluation of Next-Generation Wireless (5G) UAV Relay," *Wirel. Pers. Commun.*, pp. 1–16, 2020.
- [16] S. K. Khan, M. Farasat, U. Naseem, and F. Ali, "Link-level Performance Modelling for Next-Generation UAV Relay with Millimetre-Wave Simultaneously in Access and Backhaul," *Indian J. Sci. Technol.*, vol. 12, no. 39, pp. 1–9, 2019.
- [17] M. Zaryouli and M. Ezziyani, "Dynamic Adaptation and Automatic Execution of Services According to Ubiquitous Computing," in *International Conference on Advanced Intelligent Systems for Sustainable Development*, 2018, pp. 984–990.
- [18] G. S. Kumar, A. M. Nancy, S. Soral, and A. Shrivastava, "Query Optimization in Universal Description Discovery and Integration for Effective Web Services Discovery," *J. Comput. Theor. Nanosci.*, vol. 15, no. 6–7, pp. 2420–2424, 2018.
- [19] J. Dator, "Governing the futures: dream or survival societies?," in *Jim Dator: A Noticer in Time*, Springer, 2019, pp. 395–408.
- [20] A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, K. Van Laerhoven, and W. de Velde, "Advanced interaction in context," in *International Symposium on Handheld and Ubiquitous Computing*, 1999, pp. 89–101.
- [21] C. Shin, J.-H. Hong, and A. K. Dey, "Understanding and prediction of mobile application usage for smart phones," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 173–182.
- [22] J. H. Awan, U. Naseem, and S. K. Khan, "A proposed framework for the security of Financial Systems," *Indian J. Sci. Technol.*, vol. 12, no. 21, pp. 1–8, 2019.
- [23] B. N. Schilit, D. M. Hilbert, and J. Trevor, "Context-aware communication," *IEEE Wirel. Commun.*, vol. 9, no. 5, pp. 46–54, 2002.
- [24] P. Kremer, C. Elshaug, E. Leslie, J. W. Toumbourou, G. C. Patton, and J. Williams, "Physical activity, leisure-time screen use and depression among children and young adolescents," *J. Sci. Med. Sport*, vol. 17, no. 2, pp. 183–187, 2014.
- [25] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.
- [26] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: a survey," *Futur. Gener. Comput. Syst.*, vol. 56, pp. 684–700, 2016.
- [27] J. H. Awan, S. Memon, R. A. Khan, A. Q. Noonari, Z. Hussain, and M. Usman, "Security strategies to overcome cyber measures, factors and barriers," *Eng. Sci. Technol. Int. Res. J.*, vol. 1, no. 1, pp. 51–58, 2017.
- [28] J. Awan, S. Memon, Z. Bhutto, and W. Awan, "Jamshoro City Based Location Service (JCLS)," *J. Inf. Commun. Technol.*, vol. 10, no. 1, pp. 34–40, 2017.
- [29] J. M. Curry, "A Web of Drones: A 2040 Strategy to Reduce the United States Dependence on Space Based Capabilities," 2015.
- [30] L. Yang, J. Cao, W. Zhu, and S. Tang, "Accurate and efficient object tracking based on passive RFID," *IEEE Trans. Mob. Comput.*, vol. 14, no. 11, pp. 2188–2200, 2015.
- [31] J. H. Awan, A. Q. Noonari, R. A. Khan, K. H. Muhammadani, A. Shoro, and W. Ahmad, "Smart Phone Based Tracking System for the Billing Accuracy in Electricity Meters," *Eng. Sci. Technol. Int. Res. J.*, vol. 1, no. 3, pp. 33–38, 2017.
- [32] S. Alletto, R. Cucchiara, G. Del Fiore, L. Mainetti, V. Mighali, L. Patrono, and G. Serra, "An Indoor Location-Aware System for an IoT-Based Smart Museum," *IEEE Internet Things J.*, vol. 3, no. 2, pp. 244–253, 2016.
- [33] A. Pandharipande and D. Caicedo, "Smart indoor lighting systems with luminaire-based sensing: A review of lighting control approaches," *Energy Build.*, vol. 104, pp. 369–377, 2015.
- [34] M. R. Rahimi, J. Ren, C. H. Liu, A. V. Vasilakos, and N. Venkatasubramanian, "Mobile cloud computing: A survey, state of art and future directions," *Mob. Networks Appl.*, vol. 19, no. 2, pp. 133–143, 2014.
- [35] A. Etkin, C. Büchel, and J. J. Gross, "The neural bases of emotion regulation," *Nat. Rev. Neurosci.*, vol. 16, no. 11, p. 693, 2015.
- [36] S. A. Memon and J. H. Awan, "Transformation towards Cyber Democracy: A study on Contemporary Policies, Practices and Adoption Challenges for Pakistan," in *Handbook of Cyber-Development, Cyber-Democracy and Cyber-Defense*, 2017, pp. 1–20.
- [37] Y. J. Lee, J. Trevathan, I. Atkinson, and W. Read, "An intelligent agent system for managing heterogeneous sensors in dispersed and disparate wireless sensor network," *Int. J. Sens. Networks*, vol. 27, no. 3, pp. 149–162, 2018.
- [38] A. Mankee, M. S. Safiullah, and H. Raza, "Intelligent air-interface congestion manager." Google Patents, 2015.
- [39] M. Umlauf, C. Raffelsberger, A. Kercek, A. Almer, T. Schnabel, P. Luley, and S. Ladstaetter, "A communication and multi-sensor solution to support dynamic generation of a situational picture," in *Information and Communication Technologies for Disaster Management (ICT-DM)*, 2016 3rd International Conference on, 2016, pp. 1–7.
- [40] Ö. Yürür, C. H. Liu, Z. Sheng, V. C. M. Leung, W. Moreno, and K. K. Leung, "Context-awareness for mobile sensing: A survey and future directions," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 1, pp. 68–93, 2016.
- [41] D. Ferreira, V. Kostakos, and A. K. Dey, "AWARE: mobile context instrumentation framework," *Front. ICT*, vol. 2, p. 6, 2015.
- [42] J. H. Awan, S. Memon, S. M. Pathan, M. Usman, R. A. Khan, S. Abbasi, A. Q. Noonari, and Z. Hussain, "A user friendly security framework for the protection of confidential information," *Int. J. Comput. Sci. Netw. Secur.*, vol. 17, no. 04, pp. 215–223, 2017.
- [43] U. Naseem, I. Razzak, and I. A. Hameed, "Deep Context-Aware Embedding for Abusive and Hate Speech detection on Twitter," *Aust. J. Intell. Inf. Process. Syst.*, p. 69, 2019.
- [44] L. Snidaro, J. García, and J. Llinas, "Context-based information fusion: a survey and discussion," *Inf. Fusion*, vol. 25, pp. 16–31, 2015.
- [45] M. S. Birrittella, M. Debbage, R. Huggahalli, J. Kunz, T. Lovett, T. Rimmer, K. D. Underwood, and R. C. Zak, "Intel<sup>®</sup> Omni-path architecture: Enabling scalable, high performance fabrics," in *High-Performance Interconnects (HOTI)*, 2015 IEEE 23rd Annual Symposium on, 2015, pp. 1–9.
- [46] E. Ibarra, A. Antonopoulos, E. Kartsakli, and C. Verikoukis, "HEH-BMAC: Hybrid polling MAC protocol for WBANs operated by human energy harvesting," *Telecommun. Syst.*, vol. 58, no. 2, pp. 111–124, 2015.
- [47] N. Nebhani, J.-C. Lapayre, and M. S. Bouhleh, "Ontology traceability for the adaptation of services in pervasive environment," in *Systems, Man, and Cybernetics (SMC)*, 2016 IEEE International Conference on, 2016, pp. 4543–4548.
- [48] G. Liang and J. Cao, "Social context-aware middleware: A survey," *Pervasive Mob. Comput.*, vol. 17, pp. 207–219, 2015.
- [49] X. Li, M. Eckert, J.-F. Martinez, and G. Rubio, "Context aware middleware architectures: survey and challenges," *Sensors*, vol. 15, no. 8, pp. 20570–20607, 2015.
- [50] G.-Z. Liu, F.-R. Kuo, Y.-R. Shi, and Y.-W. Chen, "Dedicated design and usability of a context-aware ubiquitous learning environment for developing receptive language skills: a case study," *Int. J. Mob. Learn. Organ.*, vol. 9, no. 1, pp. 49–65, 2015.
- [51] M. Satyanarayanan, "Pervasive computing: Vision and challenges," *IEEE Pers. Commun.*, vol. 8, no. 4, pp. 10–17, 2001.
- [52] M. Conti, A. Passarella, and S. K. Das, "The Internet of People (IoP): A new wave in pervasive mobile computing," *Pervasive Mob. Comput.*, vol. 41, pp. 1–27, 2017.

- [53] Y.-H. Feng, T.-H. Teng, and A.-H. Tan, "Modelling situation awareness for Context-aware Decision Support," *Expert Syst. Appl.*, vol. 36, no. 1, pp. 455–463, 2009.
- [54] A. I. Maarala, X. Su, and J. Riekkii, "Semantic reasoning for context-aware Internet of Things applications," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 461–473, 2017.
- [55] I. Elhart, M. Langheinrich, N. Memarovic, and E. Rubegni, "A good balance of costs and benefits: convincing a university administration to support the installation of an interactive multi-application display system on campus," in *Proceedings of the 5th ACM International Symposium on Pervasive Displays*, 2016, pp. 197–203.
- [56] J. H. Awan, S. Memon, M. Shah, and F. H. Awan, "Security of eGovernment Services and Challenges in Pakistan," in *SAI Computing*, 2016, pp. 1082–1085.
- [57] X. Li, M. Eckert, J. Martinez, and G. Rubio, "Context Aware Middleware Architectures: Survey and Challenges," *Sensors*, vol. 15, pp. 20570–20607, 2015.
- [58] A. Forkan, I. Khalil, and Z. Tari, "CoCaMAAL: A cloud-oriented context-aware middleware in ambient assisted living," *Futur. Gener. Comput. Syst.*, vol. 35, pp. 114–127, 2014.
- [59] A. Forkan, I. Khalil, A. Ibaida, and Z. Tari, "Bdcam: Big data for context-aware monitoring-a personalized knowledge discovery framework for assisted healthcare," *IEEE Trans. cloud Comput.*, 2015.
- [60] A. Forkan, I. Khalil, and Z. Tari, "CoCaMAAL: A cloud-oriented context-aware middleware in ambient assisted living," *Futur. Gener. Comput. Syst.*, vol. 35, pp. 114–127, 2014.
- [61] G. Wu, H. Zhang, M. Qiu, Z. Ming, J. Li, and X. Qin, "A decentralized approach for mining event correlations in distributed system monitoring," *J. Parallel Distrib. Comput.*, vol. 73, no. 3, pp. 330–340, 2013.
- [62] J. Awan and S. Memon, "Threats of Cyber Security and Challenges for Pakistan," in *11th International Conference on Cyber Warfare and Security: ICCWS - 2016*, Boston USA, 2016, p. 425.
- [63] M. A. El Khaddar and M. Boulmalf, "Smartphone: the ultimate IoT and IoE device," in *Smartphones from an Applied Research Perspective*, InTech, 2017.
- [64] M. A. El Khaddar, M. Chraibi, H. Harroud, M. Boulmalf, M. Elkoutbi, and A. Maach, "A policy-based middleware for context-aware pervasive computing," *Int. J. Pervasive Comput. Commun.*, vol. 11, no. 1, pp. 43–68, 2015.
- [65] E. J. Y. Wei and A. T. S. Chan, "CAMPUS: A middleware for automated context-aware adaptation decision making at run time," *Pervasive Mob. Comput.*, vol. 9, no. 1, pp. 35–56, 2013.
- [66] K. Hisazumi, T. Nakanishi, T. Kitasuka, A. Fukuda, and others, "CAMPUS: A context-aware middleware," in *The 2nd CREST Workshop on Advanced Computing and Communicating Techniques for Wearable Information Playing*, 2003.
- [67] Á. Fides-Valero, M. Freddi, F. Furfari, and M.-R. Tazari, "The PERSONA framework for supporting context-awareness in open distributed systems," in *European Conference on Ambient Intelligence*, 2008, pp. 91–108.
- [68] A. H. Celdrán, F. J. G. Clemente, M. G. Pérez, and G. M. Pérez, "SeCoMan: A Semantic-Aware Policy Framework for Developing Privacy-Preserving and Context-Aware Smart Applications.," *IEEE Syst. J.*, vol. 10, no. 3, pp. 1111–1124, 2016.
- [69] A. R. M. Forkan and W. Hu, "A context-aware, predictive and protective approach for wellness monitoring of cardiac patients," in *Computing in Cardiology Conference (CinC)*, 2016, 2016, pp. 369–372.
- [70] M. E. Ajana, H. Harroud, M. Boulmalf, and H. Hamam, "FlexRFID: A flexible middleware for RFID applications development," in *Wireless and Optical Communications Networks*, 2009. WOCN'09. IFIP International Conference on, 2009, pp. 1–5.
- [71] A. Sengupta and S. Z. Schiller, "FlexRFID: A design, development and deployment framework for RFID-based business applications," *Inf. Syst. Front.*, vol. 12, no. 5, pp. 551–562, 2010.