ISSN: 2321-7782 (Online)
e-ISJN: A4372-3114
Impact Factor: 6.047

Volume 5, Issue 7, July 2017

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

IoT-New Trends in Middleware Technologies

Anusha R[1]
Asst Professor
Department of CSE
St.Peter's Engineering College
Hyderabad, Telangana – India

Anjaiah Adepu[2]
Asst Professor
Department of CSE
St.Peter's Engineering College
Hyderabad, Telangana – India

Abstract: ***Internet of things is an ultra large network of things communicating with each other. These things include physical devices like vehicles, buildings, smart phones, along with electronic devices like sensors, actuators, networks. These devices will move in and out with time. Thus numerous events are created and the communication between these events is a challenging task. The middlewares can help in integrating these devices, their heterogeneous computing, and interoperability within the services provided. There are a number of proposals on middlewares for IoT based on WSNs. Here we discuss about the requirements of IoT middleware, the challenges faced by them, various middleware solutions and their future research.***

Keywords: ***Internet of Things, IoT Middlewares, and Event Based Middlewares, Service Oriented Middleware, Agent Based Middleware, Database Oriented Middlewares.***

I. INTRODUCTION

With the progress of numerous technologies including sensors, embedded systems and cloud computing, along with the emergence of new cheap, small wireless devices, our daily lives have become wirelessly interoperable. By enabling easy access of and interaction with a wide variety of physical devices such as, home appliances, surveillance cameras, vehicles, machine etc, IoT leads to the development of applications like home automation, industrial automation, traffic management, medical aids etc. The huge network and the events generated by them bring new challenges in developing applications. The middle ware can alleviate the development of application by integrating heterogeneous devices, support interoperability and provide various services through these devices. Wireless sensor networks (WSNs), RFID, machine-to machine (M2M) communications, and supervisory control and data acquisition (SCADA) are the four essential components of IoT [1], [2]. The IoT middleware integrate these technologies to support the domains. In this paper we identify the key characteristics of IoT, and the requirements of IoT's middleware, comprehensive review of the existing middleware systems focusing on current, state-of-the-art and outlines open research challenges, recommending future research directions.

II. IOT AND ITS CHARACTERISTICS

Research of the IoT is still in its early stage. IoT can be viewed from three perspectives: 1) Internet-oriented; 2) things-oriented (sensors or smart things); and 3) semantic-oriented (knowledge) [3]. Also, the IoT can be viewed as either supporting consumers (human) or industrial applications.

The first definition of the IoT was from a "things-oriented" perspective, where RFID tags were considered as things [6]. According to the RFID community, IoT can be defined as, "The worldwide network of interconnected objects uniquely addressable based on standard communication protocols". European research cluster of IoT (IERC) definition, where "The Internet of Things allows people and things to be connected anytime, anyplace, with anything and anyone, ideally using any

© 2017, IJARCSMS All Rights Reserved

path/network, and any service".Companies like IBM, CISCO are using Industrial IoT to improve their Production by reducing the machine downtime which further reduces energy costs.[1], [4], [5]. Industrial IoT uses sensors, automatic communication and decision making with advanced analytics.

The definition of things in IoT includes variety of elements like smart phones, tablets which we carry around. They also include home, work place, industries with robots, machines, devices such as RFID. Enormous number of devices will be connected to internet, providing data, information and services to form IoT.

Sensor networks including wireless sensor network, wireless sensor and actuator network, RFID, M2M communication, SCADA are the essential components of IoT. The characteristics are inherited from one or more of these components. The main characteristics are presented from application and infrastructure perspectives.

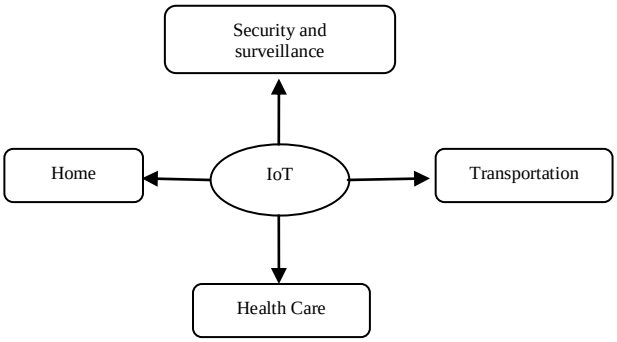

Fig I: IoT applications

III. Characteristics of IoT Infrastructure

Heterogeneous devices: The heterogeneous nature of IoT is due to the involvement of devices of various capacity and features, multi-vendor applications, etc. The IoT devices range from low power radios to reduce the power consumption, embedded and sensor network, to high computing devices performing routing, data processing etc.

Resource Constrained: The devices used in IoT environment ranges from RFID which has limited memory and processing power hence less form factor. But the devices like SCADA is having higher form factor.

Spontaneous Reaction: As the devices move around, spontaneous interaction happens when one object comes in the communication range of other object. This happens without much human interaction.

Ultra-large-scale network and large number of events: IoT will be an ultra large scale network with thousands of devices interacting to produce enormous number of events. This may lead to event congestion and reduce event processing capability.

Dynamic network: IoT contains dynamic network in which devices are connected wirelessly and will move in or out of the network at any time. So it is difficult to maintain a stable network for IoT. The nodes cooperate themselves to keep the network active and connected.

Context-awareness: Context-awareness is important factor in the behaviour of the things in the IoT [2], [6]. IoT includes large number of sensors which produce huge data which can be understood after analysing and interpreting. The context aware computing stores these data for further interpretation.

Intelligence: According to Intel, intelligent devices and intelligent systems are the key elements of IoT. These elements along with web services, SOA components and virtual objects interact and act independently based on the circumstance and environment.

© 2017, IJARCSMS All Rights Reserved

Location-aware: Location information about the objects and sensors plays an important role in context-aware computing which is an important aspect of IoT. In an ultra large scale network the interaction depends on the location of the things, other devices and the surrounding.

Distributed: The huge spatial distribution of the things in IoT makes it distributed at different scale.

IV. MIDDLEWARE IN IOT AND ITS REQUIREMENTS

A middleware aids in abstracting the hardware complexities so that the developer can focus on the problem solution. This provides a software layer between applications, the operating system and the network communications layers and coordinates the processing. The middlewares reduces the gap between the communication technologies and the system level technologies for the smooth functioning of the system. The following are the middleware requirements based on the IoT characteristics. They are further divided into 2 based on the services the middleware provide and the architecture they are supporting.

A. *Middleware Service Requirements:*

The Middleware service requirement of IoT can be functional or non functional. Functional requirement of an IoT are the services like abstraction, resource management and non functional requirements are those that provide QoS support or performance issue. The following are functional requirements.

Resource discovery: Heterogeneous IoT resources include RFID devices, analog to digital convertors, network devices, etc and the services provided by these devices. These devices announce their presence and the services they offer. Efficient distribution of load is achieved through various discovery mechanisms.

Resource management: An acceptable QoS is expected from all devices in IoT. Thus the devices must be allocated in an efficient way and resolve the resource conflicts. Thus the middleware speeds up the spontaneous resource allocation to fulfil the needs.

Data Management: The data in IoT are the sensed data, network information which are required for the application operation. IoT provide data acquisition, filtering, compression aggregation and storage services to the applications involved.

Event Management: Huge number of events is generated in IoT environment. These middlewares analyse the real time data and provide it for the application accurately so that they work intelligently.

Following are the key non-functional requirements:

Scalability: As IoT involves the entry and exit of things frequently, the system must be readily scalble. This is achieved by using IPv6 for addressing [68].Virtualization improves scalability by hiding the underlying hardware.

Reliability: The middleware must be operational till the completion of the event. This is important to achieve overall communication, data and devices reliability, the failure of which can result in unexpected output which can be even dangerous.

Availability: The middleware supporting an IoT event must be always available. The recovery time and failure frequency must be very small. This assures fault tolerance of the system.

Security and privacy: This is a critical operation of IoT. Security must be provided in every block which involves user level application. The middleware blocks using using the personal information of users must be always private.

Ease of deployment: The middlewares of IoT must be easily deployable. Complicated installation process must be avoided.

Popularity: IoT middlewares must be continuously improved by researches and developers so that it will reach many users.

B. *Architectural requirements:*

This provides program abstraction which support application developers.

© 2017, IJARCSMS All Rights Reserved

Programming abstraction: Every middleware must provide an API for applicaton developer.The high level programming interface isolates the lower IoT infrastructure.The level of abstraction deals with how the programmer views the system. Programming paradigm deals with the model of developing services.

Interoperable: the middleware must be able to work with heterogeneous devices and technologies. A network must be able communicate with othrs using different technologies.It must be able toexchange information with the growing and changing set of devices in IoT

Service based: A middleware must provide high flexibility in adding new functions. These middlewares provide high level abstraction of underlying hardware on the basis of security, reliability, data management etc.

Adaptive: In IoT the environment changes frequently. Thus the middleware must dynamically adapt easily with the changing environment.

Context-aware: The IoT environments are subject to frequent changes. Thus the context also changes. So to build adaptive systems, the middlewares must be aware of the context of users, devices and hence provide appropriate services to the users.

Autonomous: This means IoT environment are self-governed. The devices and applications communicate among themselves without any human interaction. This intelligence is achieved by the use of autonomous agents, embedded intelligence, and predictive approaches in middlewares.

Distributed: The IoT users and devices are geographically distributed. Thus the middleware must support functions distributed across the infrastructure.

V. Overview of Existing Work

The middleware solutions are highly diverse in their designs. They are grouped based on their design approaches. The following are the different middleware design groups:

1) Event-Based; 2) Service-Oriented; 3) Agent-Based; 4) Database-Oriented.

Some middlewares use hybrid approaches also which perform better than individual categories.

A. Event-Based Middlewares:

In event based middleware, the interaction among the components, applications etc are through events. The event parameters describe the changes to be made in the states. Events are sent from the sender to the receiver.

This model is based on publish/subscribe pattern. Every event will have a particular publisher whose data stream can be accessed by subscribers of that event. Notifications about the event are send to the subscribers.

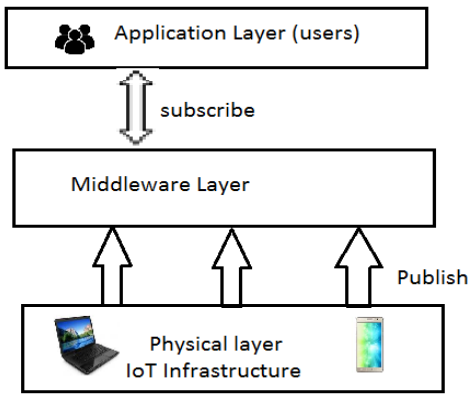

Fig 2: Event Based design model

HERMES[7] is an event based middleware used for large distributed applications. It is either type based or event based systems. It offers interoperability scalability and reliability requirements with the help of scalable routing algorithm and fault tolerant mechanisms. The following are the different layers of HERMES: the middleware layer, event-based layer, type-based and attribute-based pub/sub layer, overlay routing network layer, and network layer. This provides API for the programmers. It also provides functionality like fault tolerance, event discovery, security etc. It does not support persistent storage of events and adaptation is limited to network layer.

GREEN[8] is another event based middleware which supports highly pervasive computing applications in heterogeneous environment. It can be reprogrammed in heterogeneous environment easily and the dynamic behavior is achieved through lightweight component structure. It is less adaptable and less support for interoperability.

PRISMA[9] is a resource oriented event based middleware for WSN, with high level standardized interface for data access, interoperability with heterogeneous technologies. It has a 3 layered architecture: access layer, service layer, application layer. The access layer manages communication, data acquisition, QoS verification etc. Service layer provides resource discovery, application layer provides programming abstraction, receive and manage messages. It does not support hard real time and dynamic adaptation.

The advantages of event based systems are strong decoupling of producers and subscribers. But the interoperability, timeliness, adaptability are not highly achieved. Protocols for security and privacy need to be developed.

The event based middleware are used in systems which have mobility and failures in common. Timeliness, interoperability, adaptability are not well addressed. Appropriate protocols and models for security and privacy needs to be developed.

B. Service-Oriented Middlewares:

Service-Oriented computing is based on service oriented architecture. Even though the loose coupling, service reusability and other features of SOC makes it beneficial for IoT. The ultra large scale network and resource mobility makes use of these challenging. This can be achieved by SOM with the functionalities of publishing, discovering accessing services at run time. The service oriented middlewares can be categorised as standalone SOM for IoT or Platform as a Service model of cloud computing.

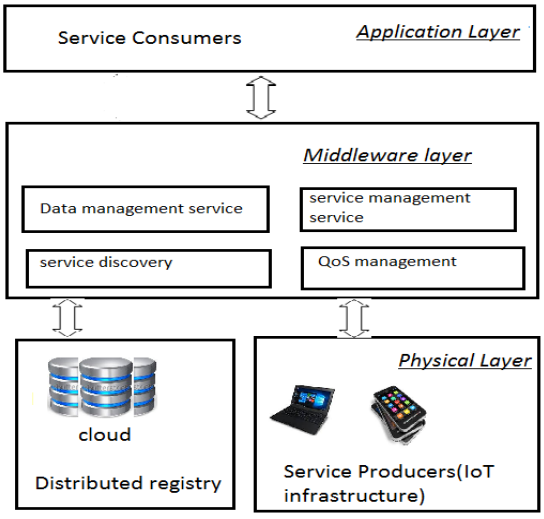

Fig 3: Service Oriented Middleware Model

HYDRA[10] or Linksmart is a middleware for ambient intelligent systems and services. It contains many components like service manager, event manager, device manager, storage manager, context manager. These are grouped into application and device elements which has a semantic layer, service layer, network layer, and security layer. It provides semantic and syntactical interoperability, dynamic and self-reconfiguration. The resource, policy and device managers optimize the energy

consumption in resource constrained devices. Security within devices during communication is achieved by security components.

The MUSIC[11] middleware provides a self-adaptive architecture for SOA environments where dynamic changes can occur in service provider and service consumer context. The context manager, QoS manager, adaptation manager, plan repository, service discovery supports QoS awareness and context based dynamic adaptation. As the contexts can contain sensitive private data, the risks of security issues are high.

SOCRADES[12] simplifies the management of underlying devices. It contains a layer for application services and another for device services. The application service layer component performs event management and storage. The service discovery component of device service layer allows to discover the services provided by real world things, the device manager deals with resource management. The role based access control authenticates the devices communicating to the middleware and backend services.

The existing SOMs do not consider abstraction and code management for ultra large IoT systems. The resource discovery and management doesn't scale good for ultra large IoTs. Storage and security trust is limited in these IoT middlewares. Thus a global scalable, secure IoT services must be developed.

C. Database-Oriented Middlewares:

The network is viewed as a Virtual Relational Database in which users can query them using a SQL like query languages. Research is done on developing distributed database for interoperating system.

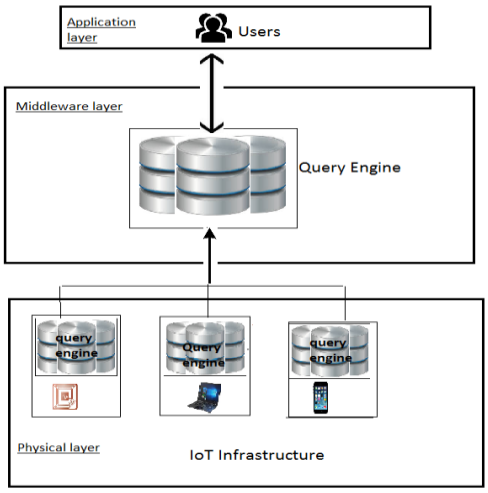

Fig 4: Database Oriented Middleware Model

SINA[13] allows the sensor networks to generate query, collect the results, monitor the network changes. It also helps in resource management. The SINA modules aids in adaptability of sensor information facilitate query, event monitoring etc. The clustered sensor nodes provide scalability and energy efficient. The interoperability, context awareness, and privacy are few issues to be resolved.

SENSATION[14] is a Database oriented middleware supports different sensors, network infrastructure, middleware technologies are used for WSNs. It provides a good programming model for context aware models. It collects requests and provide the response in real time. Event driven programming is used to trigger actions for the events generated in WSNs.

Kspot+[15] is a data centric distributed middleware architecture. It is an en source middleware that can be used for environmental monitoring, health monitoring, urban monitoring etc. Several challenges are faced, but scalability have been considered to maintain the QoS standard of Kspot+ regardless the growing networks. It does not support privacy, real time, availability, modularity etc.

The database middleware views the network as a virtual database. Easy to interfaces are available to interact with network. It provides good programming abstraction and data management support. Energy consumption is less as data is collected from

© 2017, IJARCSMS All Rights Reserved

individual nodes. Being a centralized approach, it is hard to handle large scale sensor networks. As they do not support hard real time systems, it can't be used for safety critical systems.

D. Agent Based middleware:

This is based on modular programming approach which distributes mobile agents into the network. These agents maintain their execution states while moving from one node to another. Using mobile agents has a lot of advantages like resource management, code management, availability reliability, adaptiveness and heterogeneity. They are useful in resource constrained devices.

AGILLA[16] is an agent based middleware. It reduces the code size and provides self adaptiveness. These autonomous agents provide scalability, consistency, adapt itself to changing environment. This is very useful in resource constrained devices. The challenges faced are programmability, code management and message loss in mobile agents.

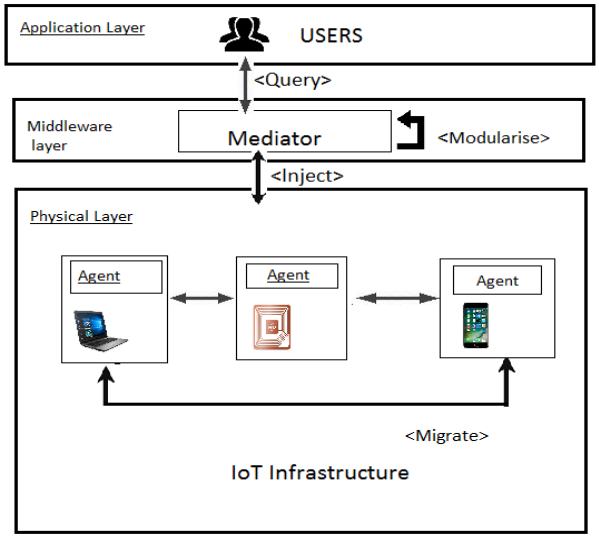

Fig 5: Agent Based MiddleWare Model

UBIWARE[17] middleware is popular in research and development because of the automatic resource discovery, monitoring, innovation etc. The UBIWARE agent has 3 layers:1) The behavioural layer based on Java, 2)middle layer with agents behavior model, 3) Reusable resource layer with sensors, rfids etc. The security policy supports security. But the support for interoperability is limited.

MAPS and TinyMAPS[18] are agent based Java oriented middleware. They are based on lightweight architecture and offer many services for agent management. The TinyMAPS is working to improve communication and migration mechanisms.

These agent based devices do not address the heterogeneous nature of IoT infrastructure. They are designed for WSNs or mobile devices. They don't provide real time solution and also some security and privacy issues. The mobile agents are also susceptible to data loss. Still these agent based middlewares can reduce design complexity by the use of high level policies.

The following table gives a brief description about the functional requirements of the middlewares discussed above.

TABLE 1 SUPPORTED FUNCTIONAL REQUIREMENTS BY IOT MIDDLEWARES

FUNCTIONAL REQUIREMENTS					
	Resource discovery	Resource management	Data management	Event management	Code management
Event Based Middleware					
HERMES	CD-DD	RM	DPF	LS	NI
GREEN	DD-ND	RM	DS,DPF	LS	CA
PRISMA	CD-DD	RM	DS,DPA	SS	CA
Service Oriented Middleware					
HYDRA	DD-DeD, DD-SD	RA,RM,RCP	DS	SS	NS
MUSIC	DD-SD	RA,RM,RCA	NS	NS	NS
SOCRADES	DD-DeD, DD-SD	RA,RM,RCP	NI	LS	NS

© 2017, IJARCSMS All Rights Reserved

Database-Oriented Middlewares					
SINA	DD-DeD	RM	DS,DPA	SS	NS
SENSATION	CD-DeD	RM	DS,DPA	SS	NS
KSPOT$^+$	DD-SD	RM	DS,DPA	NS	NS
Agent Based middleware					
AGILLA	DD-DeD	RA,RM,RCA	DPA	LS	CA,CM
UBIWARE	DD-DeD, DD-SD	RA,RM,RCA	DPA	LS	CA,CM
MAPS/TINYMAPS	DD-DeD	RA,RM/RCA	DPA/DPA,DPF	LS	CA,CM
Abbr used:	CD: Centralized Discovery DD: Distributed Discovery ND: Network Discovery DeD: Device Discovery SD: Service Discovery	RA: Resource Allocation RM: Resource Monitoring RCP: Resource Composition Predefined RCA Resource Composition Adaptive	DS: Data Storage DP: Data Pre-processing • A: aggregation • F:filtering NS: Not supported NI: No information	SS: Small scale LS: Large scale	CA:Code allocation CM:Code migration

The following table gives a brief description about the non-functional requirements of the middlewares.

TABLE 2 SUPPORTED NON-FUNCTIONAL REQUIREMENTS BY IOT MIDDLEWARES

NON-FUNCTIONAL REQUIREMENTS							
	Scalability	Security	Availability	Reliability	Real-Time	Privacy	Popularity
Event Based Middlewares							
HERMES	AL,NLIoTS	NI	S	CR,DR	HRT	NI	M
GREEN	NLIoTS	NI	S	NI	SRT	NI	M
PRISMA	AL,NLIoTS	NI	S	CR	SRT	NI	H
Service Oriented Middlewares							
HYDRA	AL, NL,WSNS	S	NI	NI	SRT	NS	H
MUSIC	AL, NL,WSNS	NS	S	DR	NI	NS	M
SOCRADES	AL,NL,IoTS	C	NI	NI	SRT	NS	H
Database-Oriented Middlewares							
SINA	NL WSN S	NS	NS	NS	NRT	NS	M
SENSATION	NLWSNS	NS	S	NI	NRT	NS	L
KSPOT$^+$	NLWSNS	I	S	NI	NRT	NS	L
Agent Based middlewares							
AGILLA	NLWSNS	NS	S	DR	SRT	NS	M
UBIWARE	AL,NL,IoTS	C,A	S	NI	SRT	NS	H
MAPS/ TINYMAPS	AL, NL,WSNS	NI	NI/S	CR/CR,DR	SRT	NI	M
Abbr Used S:Supported NS:Not supported	AL:Application level NL:Network level WSNS: WSN Scale IoTS IoT: Scale	C:Confidentiality A: Availability I: Integrity		C:Communication D: Data	H:Hard real time S: Soft real time N:Non real time	S:Supported NS:Not supported	M:Medium H:High L:Low

The following table gives a brief description about the Architectural Requirements of the middlewares.

TABLE 3 SUPPORTED ARCHITECTURAL REQUIREMENTS BY IOT MIDDLEWARES

ARCHITECTURAL REQUIREMENTS								
	Abstraction	Interoperable	Context Aware	Autonomous	Adaptive	Service Based	Light Weight	Distributed
Event Based Middleware								
HERMES	S	NeI, SeI	N	Y	DA	Y	M	Y
GREEN	S	NeI	Y	N	DA	Y	M	Y
PRISMA	S	NeI	Y	N	SA	Y	E	Y
Service Oriented Middleware								
HYDRA	S	NeI, SeI, SI	Y	NI	DA	Y	E	Y
MUSIC	S	NeI	Y	Y	DA	Y	N	Y
SOCRADES	S	NeI	Y	Y	DA	Y	NI	Y
Database-Oriented Middlewares								
SINA	S	NS	N	Y	DA	N	E	Y
SENSATION	S	SI	Y	Y	NS	NI	NI	Y
KSPOT$^+$	S	NeI	N	Y	NS	NI	E	Y
Agent Based middleware								
AGILLA	S	NeI, SeI, SI	Y	Y	DA	Y	M	Y
UBIWARE	S	NeI, SeI, SI	Y	Y	DA	Y	NI	Y
MAPS/	S	NeI, SeI	N\Y	Y	DA	Y	M/E,M	Y

© 2017, IJARCSMS All Rights Reserved

TINYMAPS								
	S:Supported NS:Not supported	NeI: Network SeI: Semantic SI: Syntactic	Y:Yes N:No	Y:Yes N:No NI: No information	DA: Dynamic SA: Static	Y:Yes N:No	E:Energy M:Memory	Y:Yes N:No

VI. Future Research Directions

In this paper we have seen the different middlewares available and their strength. Still there are challenges in the areas of scalability, availability, heterogeneous resource discovery, context awareness etc. These middlewares cope up easily with WSNs but M2M, RFID, SCADA are rarely dealt with. It indicates even though with so many advanced solutions, still these open challenges have to be addressed.

A. Challenges In Functional Requirement:

Resource discovery: The ultra large scale IoT network does not support the use of centralized registry and resource discovery methods. The registries that work under normal conditions do not scale well in the IoT environment and few others may not provide guaranteed discovery of resources. Thus further research is required to make accurate models suitable for the IoT applications.

Resource management: Frequent resource conflict occurs in IoT environment when sharing of resource occur among multiple concurrent services. So conflict resolution methods need to be implemented.

Data Management: The huge data that is collected must be aggregated and filtered to convert into usable data. The middlewares considered above provide data aggregation but not filtering. Data compression is another issue to be handled as IoT devices are resource constraint and data transmission approaches are costly.

Event management: Huge number of events is generated in IoT systems, which may create bottlenecks in the components. The above middlewares are not tested for that. Also studies must be done on how to manage complex events, continuous events etc.

Code management: This is a major challenge faced by IoT systems. Updates in the business logic must be supported by IoT. Agent specific IoT supports code management but code migration and allocation are to be worked on. Reducing the size of the interpreted code is still a challenge.

A. Challenges In Non Functional Requirements:

Scalability: Almost all the middlewares are WSN based; hence their ability to scale depends on the WSNs. Hence for the ultra large scale network, they may not be heavily scalable leading to poor performance. Scalability being a system wide requirement, middle wares components must be well scalable to achieve it.

Real Time: The IoT systems are working completely based on the physical environment. Thus real time information must be collected and processed which is a challenging task. These middlewares are mainly non real time in nature. Only very few can work with hard and soft real time services. Thus solutions must be considered for real time services and self adaptability.

Reliability: Middleware reliability can be achieved if the components can be easily replaced. This is not addressed in any proposals hence can be worked on in the future.

Availability: The failure of hardware components and their recovery are quiet common in any environment. IoT environment must be working without the unavailability of services by providing services of the failed by other components

Privacy and Security: The security trust and privacy issues in the technologies used in IoTs are not well resolved hence it exists in IoT systems also. A massive research must be done in this to achieve the security and privacy in middleware level.

© 2017, IJARCSMS All Rights Reserved

B. Challenges in Architectural Requirements:

Interoperability: Most of the IoT middlewares support network interoperability but semantic and syntactic interoperability is very limited. Few middlewares offers semantic interoperability. A research on understanding the IoT syntax and semantics is required.

Adaptive: As in most of the approaches in IoT the rules, policies, QoS definitions, are hard coded, adaptation decision requires the recompiling and redeploying a part of whole system. Research must be done to provide a flexible, dynamic and context aware adaptive model.

Context awareness and Autonomous behaviour: By exploiting of context aware resource discovery, context aware data management, the interoperability, reusability, and applicability of different components can be achieved.

VII. CONCLUSION

Middleware is important for the development of various application and things in IoT. Every middleware is different and has different requirements. In this paper, the IoT characteristics and middleware requirements are identified. A study is made on these middlewares and the issues, challenges, and research possibilities are identified. Based on the middleware design they are characterised as event based, agent based, database oriented and service oriented. They have been reviewed and the functional, non-functional and architectural requirements are described in the table. None of them cover all the requirements together. They support few requirements fully or partially. Service oriented and Agent based middlewares supports most of the requirements.

The service oriented supports abstraction and scalability. Agent based middleware is good in code and resource management. But these models face problem in providing security and privacy. The database approach is efficient for data management. But does not work in realtime approaches. Event based design have good mobility but limited context awareness, adaptability.

Although these middlewares address most of the requirements for IoT, some requirements and issues are not studied. The features like security and privacy, resource discovery, interoperability, context awareness has a good scope for future work.

References

1. H. Zhou, The Internet of Things in the Cloud: A Middleware Perspective, 1st ed. Boca Raton, FL, USA: CRC, 2012
2. C.Perera,A.B.Zaslavsky,P.Christen,andD.Georgakopoulos,"Context aware computing for the Internet of Things: A survey," IEEE Commun. Surveys Tuts., vol. 16, no. 1, pp. 414–454, May 2013
3. L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," Comput. Netw., vol. 54, no. 15, pp. 2787–2805, 2010.
4. M. Scott and R. Whitney. (2014). "The industrial Internet of Things," [Online]. Available: http://www.mcrockcapital.com/uploads/1/0/9/6/961847/mcrock_industrial_internet_of_things_report_2014.pdf
5. ECHELON. (2013). "Requirements for the industrial Internet of Things," [Online]. Available: http://media.digikey.com/Resources/Echelon/IIoTWP.PDF
6. V. Cristea, C. Dobre, and F. Pop, "Context-aware environments for the Internet of Things," in Internet of Things and Inter-Cooperative Computational Technologies for Collective Intelligence. New York, NY, USA: Springer, 2013, vol. 460, pp. 25–49.
7. P.R Pietzuch, "HERMES, a Scalable event based middleware" Univerity of Cambridge, Comput. Lab., Tech. Rep. UCAM-CL-TR-590, Jun. 2004 [Online].Available: http://www.cl.cam.ac.uk/techreports/UCAMCL-TR-590.pdf
8. T. Sivaharan, G. Blair, and G. Coulson, "Green: A configurable and reconfigurable publish-subscribe middleware for pervasive computing," in On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE. New York, NY, USA: Springer, 2005, pp. 732–749
9. J. R. Silva et al., "PRISMA: A publish-subscribe and resource-oriented middleware for wireless sensor networks," in Proc. Adv. Int. Conf. Telecommun. (AICT'14), 2014, pp. 87–97
10. M. Eisenhauer, P. Rosengren, and P. Antolin, "Hydra: A development platform for integrating wireless devices and sensors into ambient intelligence systems," in The Internet of Things. NewYork, NY, USA: Springer, 2010, pp. 367–373
11. R. Rouvoy et al., "Middleware support for self-adaptation in ubiquitous and service-oriented environments," in Software Engineering for SelfAdaptive Systems. New York, NY, USA: Springer, 2009, pp. 164–182.
12. D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the SOA-based Internet of Things: Discovery, query, selection, and on-demand provisioning of web services," IEEE Trans. Serv. Comput., vol. 3, no. 3, pp. 223–235, Jul. 2010.

© 2017, IJARCSMS All Rights Reserved

13. C.-C. Shen, C. Srisathapornphat, and C. Jaikaeo, "Sensor information networking architecture and applications," IEEE Pers. Commun., vol. 8, no. 4, pp. 52–59, Aug. 2001.
14. P. Hasiotis, G. Alyfantis, V. Tsetsos, O. Sekkas, and S. Hadjiefthymiades, "Sensation: A middleware integration platform for pervasive applications in wireless sensor networks," in Proc. Wireless Sensor Netw., 2005, pp. 366–377.
15. P. Andreou, D. Zeinalipour-Yiazti, P. Chrysanthis, and G. Samaras, "Towards a network-aware middleware for wireless sensor networks," in Proc. Int. Workshop Data Manage. Sensor Netw. (DMSN), 2011.
16. C.-L. Fok, G.-C. Roman, and C. Lu, "Agilla: A mobile agent middleware for self-adaptive wireless sensor networks," ACM Trans. Auton. Adapt. Syst. (TAAS), vol. 4, no. 3, p. 16, 2009.
17. N. Michal, K. Artem, K. Oleksiy, N. Sergiy, S. Michal, and T. Vagan, "Challenges of middleware for the Internet of Things," in Automation Control—Theory and Practice. InTech, 2009.
18. F. Aiello, G. Fortino, S. Galzarano, and A. Vittorioso, "TinyMAPS: A lightweight java-based mobile agent system for wireless sensor networks," in Intelligent Distributed Computing V. New York, NY, USA: Springer, 2012, vol. 382, pp. 161–170
19. Mohammad Abdur Razzaque, Marija Milojevic-Jevric, Andrei Palade, Siobhán Clarke, "Middleware for Internet of Things: A Survey," in IEEE Internet of Things Journal, Volume: 3, Issue: 1, Feb. 2016, pp 70 – 95.

AUTHOR(S) PROFILE

Anusha R, received B.Tech. in computer science and engineering from CUSAT, Kerala and M.Tech. in Computer Science and Engineering from JNTUH Hyderabad. She is currently working as Asst.Professor, Department of Computer Science and Engineering at St.Peter's Engineering College, Hyderabad, TS, and INDIA. She is having 5 years of Teaching experience, published few papers in National/International Journals. Her areas of research include Adhoc sensor networks, Information security, Internet of Things, Cloud computing, Manets.

Anjaiah Adepu, received B.Tech. in Computer Science And Engineering from JNTUH Hyderabad and M.Tech. in Computer Science And Engineering from JNTUH Hyderabad, He is currently working as Asst.Professor ,Department of Computer Science and Engineering at St.Peter's Engineering College, Hyderabad, TS, INDIA. He is currently working toward the Ph.D. degree with the Department of Computer Science and Engineering at Shri Venkateshwara University, Gajraula (U.P.) .He is having 12 years of Teaching experience, published more than 20 papers in National/International Journals/Conferences. He is a Member of IAENG (International association of Engineers), ISOC(Internet society of India), CSI(Computer Society of India). His areas of research include Adhoc sensor networks, Information security, Internet of Things, cloud computing.

© 2017, IJARCSMS All Rights Reserved