ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 7, January 2015

The Integrated Middleware Framework for Heterogeneous Internet of Things (IoT)

Bhagyashri Katole , Suresh V., Gita Gosavi, Amit Kudale, Gokul Thakare, Girishchandra Yendargaye, Ch. Pradeep Kumar

Abstract*— The rise of Machine-to-machine (M2M) communications is ingredient for transforming and automating our lives and society around us. A key enabler in accessing Internet of Things (IoT) resources as service end points is the M2M Application Programming Interface (API). The M2M API provides the means for the device to expose its capabilities and the services it may offer, so that remote machines may utilize them. Consequently, such APIs are necessary to enable proactive and transparent communication of devices, in order to invoke actions in IoT devices and receive the relating responses. The main idea of IoT is to expand the network extending it into homes, offices, building, cities. This leads to requirement of loosely coupling, reusability, scalability along with integrity. Now, in the era of smart object, service oriented methodology is providing efficient solutions. This paper will describe our journey to develop IoT framework starting from M2M APIs towards scalable service oriented architecture that leverages various opportunities to develop various applications using the same. The paper discusses about intrinsic characteristics of IoT with requirements of M2M APIs for IoT framework. It also talks about our M2M communication protocol stack prototype. It explains about how legacy M2M APIs are used in the IoT framework and their importance. This also explains about role of RESTful web services in service oriented framework along with challenges of creating IoT applications. It also discusses about our approach towards development of service oriented IoT framework.

***Index Terms*— Machine-to-Machine (M2M), Internet of Things (IoT), *Code division multiple access* (*CDMA*), Wideband Code Division Multiple Access (WCDMA), Service Oriented Architecture (SOA), REST, Near Field Communication (NFC), RSA, Universal Unique Identifier (UUID), Uniform Resource Identifier (URI), Application Programming Interface (API).**

I. INTRODUCTION

Internet of Things (IoT) is an integrated part of future internet and could be defined as a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocol. Some researchers think that IoT is for addressing physical objects and some think that IoT is ubiquitous with feature of "everything connected, intelligently controlled and anywhere covered". In IoT, things can be classified as into two types i.e. physical things and virtual things. The physical things include objects, behaviors, tendencies and physical events. The virtual things include entities; actions that indicate the processing of virtual things and services that are offered for certain goals.

Machine to Machine (M2M) is paradigm in which end to end communication is executed without human intervention connecting various things to IT core network. Here, things involve commercial terminals that act automatically or on remote request.

End to end communication involves network that acts as access and core network backhaul enabling connectivity and taking care of security, session management and mobility management. IT core network involves data aggregation and processing involving data caching and also its interpretation. The M2M refers to technological system that includes both wireless and wired systems to communicate with other devices of the same ability. The 'Things' in the IoT, or the 'machines' in M2M, are entities whose identity, state is capable to connect to IT infrastructure using internet.

Today, M2M is primarily being used to collect vast amounts of machine data and the IoT goes one step further by integrating data from various devices, allowing humans to intelligently interact with devices, devices with devices and devices back to humans to provide the ultimate social media collaboration of man and machine. [1]

The IoT will not only have a broad impact on our everyday life in the near future, but also create a new ecosystem involving a wide array of players such as device developers, service providers, software developers, network operators, and service users. It facilitates the entrance into the IoT related mass market, and establishing a global IoT ecosystem with the worldwide use of devices and softwares. [2] The IoT has following functional intrinsic characteristics

A. Sensing – The IoT includes all sensing technologies that realizes things identification. This includes thing's identifier (ID), static attributes (e.g. MAC, size, company etc.) and environmental information (e.g. temperature, pressure and humidity). One important goal of future IoT is to unify the physical world and cyber world and solve human machine interface bottlenecks by using sensing technologies. In addition, along with more and more things mapping from physical world to cyber world and thus control of physical world is becoming wider.

B. Network of networks – The IoT involves various kinds of networks. On the one hand, there are many heterogeneous access networks in the sensor-actuator layer. On the other hand, heterogeneous control networks such as GSM, CDMA, and WCDMA coexist in the network layer. Future IoT should construct the network of networks which is crucial part of IoT development to connect physical things and virtual things.

C. Intelligent processing – In IoT, processing should be realized intelligently. An ideal scenario is that things can be sensed and controlled automatically with high

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 7, January 2015

intelligence, real-time information management, flexible scheduling and accurate tracking. Intelligent processing capabilities and freeing human from information explosion is another goal of future IoT. [3]

A solid, well-designed M2M API provides the basis for the simplified management of resources. Additionally, the main advantage of such an API is that it provides the abstraction layer necessary to realize interactions between IoT devices uniformly. The starting point for defining the actual services for the IoT endpoints should expose via this M2M API. Towards this direction, the current state-of-the-art in M2M communications, in terms of standardization bodies, research projects, protocols, software development platforms, already deployed prototypes need to be taken into consideration.

In most telecommunication networks, transactions happen between several components or applications in a session based communication manner, as all these components are always connected to the network and in on-state. The case is different in M2M networks, which connect huge number of devices with limited storage capabilities and are more likely to face energy shortages. Such M2M devices might not be able to stay connected all the time, and thus it can't interact immediately to all transaction with other component in the network, this situation will cause cancellation of the transaction and other effects in all parties. [4] The REST based architecture provides a good solution to this problem in M2M networks, by handling transactions in a resource-based communication.

A middleware or framework supporting the communication with smart things allows development of new applications and improvement of old ones. Service orientation requires loose coupling of services with operating systems, and other technologies that underling applications. Services and their corresponding consumers communicate with each other by passing data in a well-defined, shared format, or by coordinating an activity between two or more services.[5] The Service Oriented Architecture (SOA) is highly used in business projects of IoT because of loose coupling, reusability, scalability along with integrity.

The sections II in this paper will talks about requirements of M2M APIs for IoT framework. The section III discusses about our M2M communication protocol stack prototype. The section IV explains about how M2M legacy APIs are used in the IoT framework and discusses about importance of APIs. The section V explains about the role of converging M2M APIs towards RESTful web services in IoT based service oriented architecture and challenges of creating IoT applications. The section VI talks about our prototype of service oriented architecture of IoT framework. The section VII will describe future scope and the conclusion is given in section VIII.

II. REQUIREMENTS OF M2M APIS FOR IOT ARCHITECTURE

A basic common API that can be extended in a well defined way will allow fulfilling more complicated requirements, while offering the simplicity needed in other cases. Given the heterogeneous nature of the applications and device categories targeted in the IoT, the M2M API concept needs to be adaptable to the capabilities and requirements of the specific use case. These can be categorized as

A. Requirements of communication – The system shall support event-based, periodic, and/or autonomous communication. Devices shall be remotely controlled and configured. The system shall support the real-time monitoring of the radio activity of devices and gateways. The system shall support prioritization of services. The system shall provide secure and reliable communication. The system will take care of mobility of devices along with network are some of important requirements for communication.

B. Requirements of device control - The M2M API should provide an interface for controlling the device. This includes the configuration of the device as well as support for remotely activating, deactivating or updating the device.

C. *Requirements of server and client communication models* - The API should provide support for operating a device as server and client. The event-based, autonomous (unsolicited) and periodic communication, as well as request-response communication behaviors are possible by changing the server/client role.

D. Requirements of device status monitoring - The status like battery, working status of a devices in the infrastructure need to accessible for device status.

E. Communication failure notification - The API should provide an interface to inform about failures of the communication. This information provides a handle for mechanisms ensuring communication connectivity, device failure notification, robustness and reliable communication.

F. *Device Capabilities* - The API should provide information of the capabilities of a device and prioritization among devices.

To address these challenges, the emerging IEEE 802.11ah specifications are proposing a number of improvements and new features. [6]

III. THE M2M COMMUNICATION PROTOCOL STACK

The M2M API uses the underlying protocol stack to realize communication among end devices. As M2M APIs expose the device capabilities and will also support remote device management and configuration, it is evident that security becomes a dominant issue. Furthermore, the M2M protocol stack must provide the mechanisms necessary for discovering devices that join the network, along with their capabilities. In order for the services offered by a device to be accessible, the

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 7, January 2015

device must primarily be identified. Additionally, the services it offers must be published so that they are accessible by other devices. Another important aspect of the protocol stack is its capability to differentiate message flows and prioritize datagram. Thus, it will be possible to have a M2M system that will support different categories and types of services, enabling service providers to enforce pricing and charging policies. The standardized communication protocol stack is prototyped that will receive and forward the requests from physical entities in IoT environment as shown in Fig. 1.

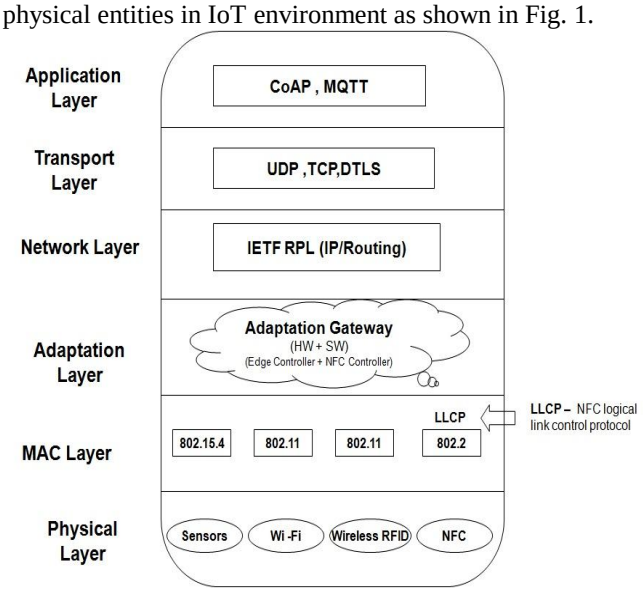

Fig 1 – Communication protocol stack

As shown in Fig. 1, the layer 2 of this protocol stack consists of MAC based on corresponding standards of devices like 802.15.4 for sensors, 802.11 for Wi-Fi, 802.2 for NFC and 802.3 (Ethernet) for RFID and so on. The layer 2 will also takes care of additional MAC layer protocols that are required for respective devices like NFC logical link control protocol (LLCP) for NFC devices. The layer 3 i.e. adaptation layer is additional important layer. This layer consist of hardware and software both involving adaptation gateway, mechanism of edge router involving communication of 802.11 and 802.15.4 and also mechanism to adapt communication among other devices. The layer 4 is for IP/ Routing. Various IPv6 based routing protocols like RPL that will also be used for low power and lossy network. In layer 5 i.e. transport layer, both UDP and TCP protocols are considered. The layer 6 consists of advanced light weight application protocols like constrained application protocol (CoAP) i.e. based on UDP and message queuing telemetry transport (MQTT) protocol i.e based on TCP. This standardized protocol stack will provide common communication platform for all devices in infrastructure of IoT.

IV. M2M APIS IN IOT FRAMEWORK

The NISG, C-DAC, Pune developed IoT framework with legacy M2M APIs having objective to facilitate inter connectivity among various device with seamless data exchange and currently targeting various domain specific applications like location aware, smart school and smart office etc. The framework will consist of device communication and discovery layer that helps in standardized communication and addressing mechanism that will help in secured communication and data exchange among various devices in IoT environment like sensors, Wi-Fi devices, NFC devices, RFID devices and also actuators. The security layer will consists of essential components of security like Authentication component, Authorization component, accounting component. These components will provide privacy, integrity, confidentiality and also availability features. The device communication and discovery layer will consists of following modules

A. ***Device Discovery*** **-** The module that provides mechanism for getting device discovered in the network of IoT framework and assigning address to the same.

B. ***Device Sensing*** **-** It involves mechanisms to sense/listen various resources and devices in IoT infrastructure so as they will able to send the data to device communication component and other components of the layer.

C. **Device *Communication*** **-** This takes cares of receiving data from various devices running on different protocols and providing intelligent information to other modules of IoT framework. This also involves publisher and subscriber model for communication.

D. ***Device Manager*** **-** This manages various devices in IoT environment in terms of maintaining information and status, managing identifiers and helps in acquiring knowledge about IoT infrastructure at the given moment. This component will also manage various physical entities/devices in IoT environment.

Multiple APIs are provided by modules serve different needs of service layer and applications. The IoT framework provides various APIs for different technologies like sensors, RFID, NFC, Wi-Fi so that services and various applications can directly use these libraries of APIs.

V. CONVERGENCE OF M2M APIS TO RESTFUL WEBSERVICES

In IoT, we already witnessed the era of connecting machine-to-machine. Today more communication service providers are opening infrastructure to 3rd party developers through open APIs. Hence API growth rate converging towards services from WEB 2.0 is increased in past few years. In M2M technology, the problem is constrained devices might not be connected all the time and thus they cannot immediately interact to all transactions in the network. REST is based on concept of resources identified by URI, hence it provides placeholder to M2M device to store their states and data. With the help of handling transactions in resource based communication, the REST based architecture provides efficient solution to the problem in M2M network.

The major challenge behind creating applications using IoT is that stakeholders are very varied with multiple and very different constraints. The service oriented approach (proposed by Services Oriented Computing, SOC) can

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 7, January 2015

facilitate this integration. Combining services is a way to easily create applications. The needs of IoT users are varied, specific to each of them (individual or organization, home automation, smart building, smart cities). IoT applications must be precisely configured according to user needs.

The "Service" approach limits dependency between stake-holders by introducing loosely coupled links. By providing an interface describing the exchange and hiding its implementation (and its possible change or modification), the dependence of the consumer to the producer is shrinking. The service is provided universally, and the specifications of related to the real hardware are hidden. Calls to the service, or calls between services, are executed in a standardized manner. [7] This facilitates interactions and limits dependencies to the mere compliance with the interface description. This approach has been successful in the Internet of Data, where no user wonders which kind of hardware or operating system is running the web service he is currently using.

VI. PROTOTYPE OF SERVICE ORIENTED IOT FRAMEWORK

The Networking and Internet Software Group (NISG) at C-DAC, Pune developed node.js based IoT framework providing various RESTful services. The "node.js" as development technology is used. The node.js is a platform for easily building fast, scalable network applications. It uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices. Various npm modules are already available and can be inherited for specific implementations. In this project, various npm modules have been used. E.g. npm-rsa for RSA implementation, npm-geolite for finding geolocation from IP, npm-uuid to generate UUID etc.

The framework uses "mongoDB" as a data-store. It is an open-source document database and the leading NoSQL database. The framework uses node.js based module "restify" to build correct REST web services. The framework provides various services that can be used by developer community as well as individual users. Developers can use the framework code to develop the applications on top of it. The framework provides web-portal so that individual users can use it for the personal applications.

As shown in Fig. 2, the Management Layer in the framework includes various services for managing projects, devices and users. The Service Organization layer will comprises of plethora of IoT services that includes implementation of various mandatory services like: registration/unregistration, subscription/unsubscription, update the status, checking status of device, finding all devices in the network, sending messages to device/devices, data retrieval services from the resource etc. The framework uses node-rules for rule engine that takes rules written in JSON format as input. This will facilitate rules service to take decision as well as logic and data separation. The framework uses RSA to secure information communication. It uses the node.js packet manager module "npm-rsa".

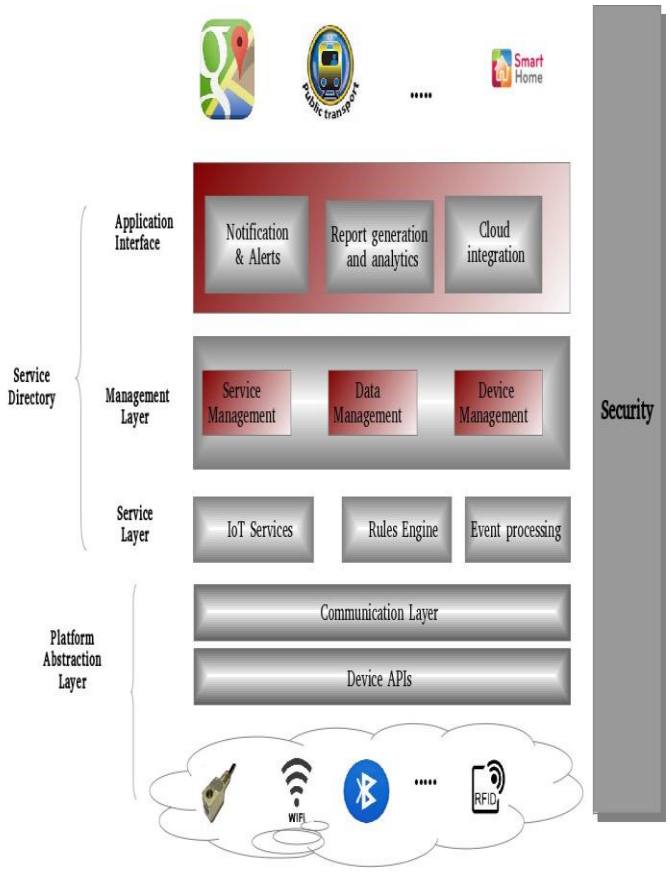

Fig 2 – Architecture of IoT framework

It provides key generation, key exchange, data signing, data verification etc. The framework also incorporated UUID concept. A node package module "npm-uuid" generates UUIDs from truly-random or pseudo-random numbers. Event processing component acts as a flexible and scalable connector among event sources (i.e. Publishers) and event consumers (i.e. Subscribers). This helps to distribute information about events of interest that will also generate alarm. Report generation and analytic component analyzes the information and generate report. The web portal provides user interface to facilitate service invocation for the user. The web portal development tools like HTML5/ CSS3, ExpressJS and HBS engine are used to provide various services to application developers and users.

A. Location Aware Services - The framework will also provide location aware services that can be used to acquire a user's location. These services have multiple purposes; on one hand, they provide an easy solution for monitoring the user's real-time location using the framework. On the other hand, service developers can use these services to create third-party location-based services based on the framework, yet those are not implemented in the framework.

The location aware services in the framework can be used for tracking a user's position in the environment that has multiple positioning devices. They also work when the technological

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 7, January 2015

circumstances of the positioning change, e.g. switching from GPS to Wi-Fi/tags/QR code. A user's position is tracked by various positioning technologies such as Wi-Fi access points, GPS, cell phone (GSM or 3G), Wi-Fi/Bluetooth/NFC tags as well as QR code. The location-data is forwarded to the framework with the help of mobile that makes the positioning technology transparent for the upper level systems.

The GPS coordinates will be taken into consideration in the outdoor scenario. When the user moves in the indoor where GPS related black spots exists (i.e. GPS is not available), the position can be still determined using mobile phone with the help of Wi-Fi access points, QR codes or RFID/NFC positioning devices. Neither the user nor the developer has to worry about the positioning technologies and devices as the location aware services in the framework will masks them from the application.

VII. CONCLUSION

The IoT envisions building a convergent platform to share dynamic data from smart objects and middleware services that process the data. The IoT framework enables IoT resource sharing at different layers of IoT service chain by converging M2M APIs in service oriented approach. It offers a way to diverge numbers of vertical M2M applications into a flat IoT service platform. The IoT framework will leverage the development of various smart applications. We believe that the IoT framework will play an important role in not only bringing the IoT ecosystem into our daily lives, but also giving various opportunities towards development of "Smart City".

VIII. FUTURE SCOPE

Today in the era of "Smart City", there is need of developments of various smart applications. The IoT framework can be effectively used to develop many applications like navigation, smart transport, smart home automation etc. The usage of IoT framework in this plethora of smart applications needs to ensure reliability. Here, reliability incorporates the issues of security, privacy, availability, robustness and flexibility to changing environmental conditions. Hence, IoT framework needs to incrementally evolve to meet requirements so that it will able to sustain along with these various concerns and upcoming challenges.

REFERENCES

[1] Internet-of-Things Architecture IOT-A Project Deliverable D3.1 - Initial M2M API Analysis.

[2] Jaeho Kim, Jang-Won Lee, "OpenIoT: An Open Service Framework for the Internet of Things", IEEE World Forum on Internet of Things (WF-IoT), pp-89-93, March 2014.

[3] Huansheng Ning, "Unit and Ubiquitous Internet of Things", CRC press, pp. 5-9, 2013.

[4] Asma Elmangoush, Thomas Magedanz, Alexander Blotny, Niklas Blum,"Design of RESTful APIs for M2M Services", 16th international conference on intelligence in next generation network, pp 50-56, Oct 2012.

[5] Yelin HONG, "A Resource-Oriented Middleware Framework for Heterogeneous Internet of Things", International Conference on Cloud Computing and Service Computing, pp 13-16, Nov 2012.

[6] Ali Hazmi, Mikko Valkama and Juho Pirskanen, "IEEE 802.11AH: promising technology for IoT and M2M applications", Internet-of-things magazine, Finland, pp. 22.

[7] Sylvain Cherrier_, Yacine M. Ghamri-Doudaney, "The "Object-as-a-Service" Paradigm" IEEE Global Information Infrastructure and Networking Symposium (GIIS), pp 1-7, Sept 2014.

AUTHOR BIOGRAPHY

Bhagyashri Katole is working as Senior Technical Officer of the Networking and Internet Software Group (NISG) at C-DAC in Pune, India and has 9+ years of experience. She completed M.S in software systems and having specialization of computer science in B.E. Her areas of interest include network security and cryptography, computer networking, emerging technologies like Internet of Things (IoT) and embedding of products on various embedded platforms. She is currently working on design and development of IoT framework along with applications.

Suresh V. is currently working as Joint Director of the Networking and Internet Software Group (NISG) at C-DAC in Pune, India. He completed his B.E and MBA. He has a strong 12+ years of experience in area of telecommunication and information technology. His area of interest includes Network Management Systems, Wireless Sensor Network, IoT and cloud computing.