Journal of the Indian Institute of Science
A Multidisciplinary Reviews Journal
ISSN: 0970-4140 Coden-JIISAD


REVIEWS

# Software Platforms for Internet of Things and M2M

*Balamuralidhar P., Prateep Misra and Arpan Pal*

**Abstract |** The technologies and applications consolidated under the vision of Internet of Things (IoT) and Machine to Machine Communications (M2M) are attracting the interest of businesses and poised to trigger next disruption in Information & Communication Technology (ICT) applications. The domain specific solutions involving sensors, mobile phones and other devices are maturing to integrate in a generic ICT services paradigm. Architecture for a unified horizontal services platform is emerging. This paper surveys important trends, key requirements, evolving technologies and emerging solutions for such a software platform for IoT and M2M services.

## 1 Introduction

The Internet of Things (IoT) explores the power of connectivity to digitally enhanced objects generally known as "smart things". The connected objects interact based on standard and interoperable communication protocols where physical and virtual "things" have their identities, physical attributes, and virtual personalities. They are seamlessly integrated into the information network using intelligent interfaces.as active participants in business, information and social processes. They are enabled to interact and communicate among themselves and with the environment. In many cases they sense the environment around them and autonomously react to the "physical world" events by triggering actions and services. Such "smart things" facilitates interaction over internet through standard Interfaces in the form of services. The functionalities provided by these 'things' can be termed as 'real-world services' as they provide a near real-time state of the physical world. A structured, machine-processible approach to provision such real-world services is needed to make heterogeneous physical objects accessible on a large scale and to integrate them with the digital world. They query and change their state and any information associated with them, taking into account security and privacy issues. In such systems applications, services, middleware components, networks, and endpoints will be structurally connected in entirely new ways and architectures.

There are two predominant perspectives in the IoT vision; one is "Things" oriented and the second is "Internet" oriented.[1] The "things" oriented visionaries look at IoT as object centric such as attaching RFID to objects,[2] wireless enabling of home appliances and so on. "Internet Oriented" approach focuses on connecting objects in an internet like network fabric. When "things" interact over "Internet" then they need the meaning of the data and actions to be represented explicitly to enable intelligent interaction and this leads to a third perspective of IoT towards "Semantic Oriented" Vision.

Semantic technologies based on machine interpretable representation have shown promise for describing objects, sharing and integrating information, and generation new knowledge through inference along with other intelligent techniques. In IoT also this is important; however, the dynamic and resource-constrained nature of the IoT requires special design considerations to be taken into account to effectively apply the semantic technologies on the real world data. For example a climate controller that uses multiple temperature and humidity sensors in a building to control the cooling environment that have different accuracy/resolution levels and report the data in different units (°C and °F ). In this case the controller needs to have sufficient meta-data information about different sensors and interpret them accordingly before decision making. In another

*TCS Innovation Labs, Tata Consultancy Services Ltd.*

case a location monitoring application infers the location of a user device through the location of W-Fi access point where the device is connected when there is a failure in accessing the GPS data. Semantic technologies enable such sensor fusion, information discovery and robustness in the IoT information processing system. Therefore, issues related to how to represent, store, interconnect, search, and organize meta data and other information generated by the IoT will become very challenging with their scale and heterogeneity. In this context, semantic technologies could play a key role. A detailed discussion on "Semantic oriented" IoT visions are available in the literature.[3–5] In fact, these can exploit appropriate modeling solutions for 'things' description, reasoning over data generated by IoT, semantic execution environments and architectures that accommodate IoT requirements and scalable storage.[3]

Today IoT is a collection of diverse technologies used to build applications in diverse domains. To abstract the diversity of underlying technologies, a software service platform is envisaged that incorporate a middleware. The middleware is a software layer or a set of sub-layers interposed between the technological and the application levels. Its feature of abstracting the details of different technologies frees an application developer from issues that are not directly pertinent to her/his focus. The middleware is gaining importance in recent years due to its major role in simplifying the development of new services and the integration of legacy technologies into new ones. A good review of IoT middle wares is available in.[6]

Due to the long-cycle market feature of IoT/M2M, most current business cases use the vertical integration model, in which the service provider should configure the service infrastructure on a case-by-case basis as the individual application demands.[7] Kouji et al.[8] argues that a horizontal integration model is more desirable than the vertical model to stimulate the market for M2M services and that, in the horizontal model, a common M2M service platform is needed to connect and collect information on various devices and to be able to manage them. Further the development of applications that integrate the functionality of multiple smart things remains a challenging task, because it requires expert knowledge of each platform. To facilitate this, recent research initiatives tried to provide uniform interfaces that create a loosely coupled ecosystem of services for smart things.[9,10] The technical challenges related to the functions or capabilities of the service platform include service enablement, device management, connection management, and M2M application deployment.

Further what is needed for a good software platform is a common means of connected application development that can leverage tools across families of interrelated devices and diverse domains. Customers will need a single unified framework to design and build solutions that can interoperate across diverse data environments and under widely differing usage scenarios. It should enable quicker development process that reduces solution development time, increases quality, fosters reusability and increase in solution development velocity.

## 2 Platform Requirements—A Perspective

In our view a platform designed for IoT applications must cater to the needs of four different types of users—namely a) End Users of IoT applications, b) Application Developers, c) Sensor providers and d) Administrators who operate and maintain the platform. The envisaged platform consists of a set of services, typically delivered over a Wide Area Network or the Internet using IP based transports such a HTTP, TCP or UDP or higher level protocols using these as the underlying transport.

### 2.1 *Platform users*

Application developers are those who register themselves to the IoT platform and use the platform services to develop and deploy applications. The applications themselves may be distributed onto both "edge" devices such as sensor nodes or gateway devices and backend cloud hosted servers. Application developers are provided necessary computing resources and Application Programming Interfaces (APIs) in order to develop applications and then deploy them on the platform. Application developers use API keys to identify and authenticate themselves to the platform and get access to services and resources as per their entitlement. We may also refer to application developers as the "tenants" of the platform.

Sensor providers are those who own and/or operate sensors and contribute sensor observations to the platform, either for their own private use or for use by others based on access control and privacy policies. Sensor providers use APIs provided by platform to push data to the platform.

Platform Administrators use the services, APIs and tools provided by the platform to manage and monitor users, services & devices. Administrators provide compute, storage and network resources to application developers and keep track of resource usage.

Finally we have end users of IoT applications. These are the public at large who consume the applications developed by application developers.

### 2.2 *Platform services*

There are different types of services need to be provisioned on the platform. They include services data management services, sensor/device management, data storage, analytics & visualization and Application & User Management.

a. ***Device Management Services*** are used by both application developers and administrators to register sensors and gateway devices, give them a unique identity, address the devices, check their health and connectivity status, install and update software on the devices and access resources and consume services from the devices.
b. ***Sensor Services*** are used by IoT applications to register sensors/observers in the platform, create meta-data about the sensors. The meta-data includes the features of the sensors and the real-world phenomena that they measure or observe, the geo-location of the sensors and the real-world entity observed. Sensor Services are also used to capture, store and query sensor observations. Sensor Services are therefore at the core of any IoT platform. Applications also need the ability to discover the sensors and their capabilities and support for this is needed from the Sensor Services. Sensor Services need to scale easily as the number of sensors and the number of observations increase. Also, they need to have the capability to handle almost any type of sensors and any kind of sensor observation. Often, the sensor observations are in the form of time series data and the platform needs to support high performance time series database.
c. ***Storage Services*** are used by application developers to store application specific data persistently. This data is typically not the sensor observation data, since that is taken care by the Sensor Services. Typically Storage Services include Relational Database services, schema free document database services, Blob and file storage facilities.
d. ***Analytics Services*** are needed to provide the platform infrastructure to perform various batch and real-time analytics on the sensor observations. This include services for Statistical processing, data mining, machine learning, aggregations and correlations, pattern detection, rule based processing, in-stream real-time analytical processing. As with all other services, Analytics Services need to be highly scalable. Complexities of the underlying infrastructure need to be hidden from the application developers and convenient APIs must be designed to enable applications to easily access the analytics platform. Moreover, it must be realized that, analytics is very problem and domain specific. Hence the actual analytics algorithms are developed as part of the applications themselves. These algorithms may be written in a variety of languages. Often scripting languages such as Python and R are used for the actual algorithms. The platform must therefore be polyglot.
e. ***Visualization*** of sensor data is another important requirement and goes hand in hand with analytics. The IoT platform must provide necessary tools and APIs for creating rich visualization of captured sensor data as well as processed data resulting from analytics.
f. ***Application and User Management*** services are used by platform administrators to create tenants on the IoT platform, provide resources and provision platform services to tenants. Application developers themselves use these services to request for resources and services and manage the life cycle of the applications.

### 2.3 *Multi-tenancy*

IoT platforms need to support multiple tenants/users in a manner such that the tenants, their data and resources are protected from other tenants on the platform. Also the common platform services must be run in such a way that tenants are not able to adversely impact their performance and availability in any way. It should be possible to scale and upgrade both common services as well as tenant specific services independently. One way to achieve these objectives is by providing tenants with separate virtual machines for their applications and database services. Also, the common platform services are run in their own separate virtual servers in their own separate virtual network domains.

### 2.4 *Integration with infrastructure-as-a-service clouds*

Integration with Infrastructure-as-a-service clouds enables IoT platforms to be highly scalable and multi-tenanted. Figure below shows how tenant specific services and common platform services may use auto-scaling systems and load balancers services to achieve scalability. These services monitor the load on each server and then create additional compute resources using the Infrastructure

cloud services and then balance the load between all the available resources.

### 2.5 *Open and sharable sensor data repository*

It is important that the IoT platform enables access to sensor configuration data and sensor observations data as a service. This platform must support virtually any kind of sensor and sensor observations. The schema and the semantics must be based on open standards and ontologies. Depending on data sharing policies set by users, sensor data may be shared either publicly or to targeted users or application developers. Application developers can query the repository for sensor data. Multiple different types of sensor observations may be combined to create different kinds of intelligent apps. Sensor data must not be locked up in separate application specific silos.

## 3 Platform Architectures—Related Research

Predominantly the software platform architectures for IoT in the literature incorporates a middleware as an abstraction layer and follows a Service Oriented Architecture (SOA) approach.[11] The adoption of the SOA principles allows decomposition of singular systems into applications consisting of a set of simpler and well-defined components. The use of common interfaces and standard protocols allows for flexibility in service composition and reduction of the time necessary to adapt to the changes imposed by the application evolution.[12,13]

Since there are no commonly accepted layer architectures for IoT, there is a difficulty in specifying a common set of services and an environment for service design and composition. An integrated architectural approach given in[14] proposes following IoT specific layers over the SoA namely a) Service Composition b) Service Management and c) Object Abstraction. Service composition provides for building specific applications through the composition of atomic services derived from various connected objects. A basic set of management services required includes: object dynamic discovery, status monitoring, and service configuration. This basic set could be extended with additional functionalities involving QoS management, semantic interoperability functions etc.[15] Object abstraction is needed for heterogeneous connected objects for harmonizing the access to the different objects/devices.

Two types of service-oriented architectures stand out as potential candidates to enable uniform interfaces to smart objects: the Representational State Transfer (REST)[16] and WS-*[17] Web services. While both the architectures provide developers with abstractions (i.e., APIs) for interacting with distributed services, they tackle loose coupling and interoperability differently. In,[18,17] REST and WS-* are compared and the authors suggest that WS-* services should be preferred for professional enterprise application integration scenarios and RESTful services for tactical, ad-hoc integration over the Web. In cases with strong security requirements, WS-* has a competitive advantage.[19,17] The WS-Security standard offers a greater number of options than HTTPS (TLS/SSL) such as encryption and authentication beyond the communication channel, endpoint-to-endpoint. Internet of Things applications pose novel requirements and challenges as neither WS-* nor RESTful Web services are able to address them effectively because they were primarily designed to be used on business or Web servers. In[20] authors reports a study to understand developer perspectives and show that the participants almost unanimously found RESTful Web services easier to learn, more intuitive and more suitable for programming IoT applications than WS-*. While we agree to the general finding of this study, in our opinion this developer friendliness has various other influencing factors including the maturity of development tools, standards and mature user communities around them.

Thiago et. al[21] presents a service oriented architecture abstracting all sensors and actuators as services in order to hide their heterogeneity, and relies heavily on a knowledge base that carries information about sensors, actuators, manufacturers, physical concepts, physical units, data models, error models, etc. It consists of three parts: a Discovery module, and Estimation & Composition module, and a Knowledge Base (KB). They use probabilistic discovery, approximately-optimal composition and automated estimation to address the massive scale and deep heterogeneity of the IoT system.

In[22] the authors present a software platform designed to facilitate the development of new IoT services based on Ubiquitous Sensor Network (USN)[23] architecture. It follows the design principles of unified information modeling, unified communication protocol and a horizontally layered approach. The architecture uses USN gateways at access layer to provide the unified entry point for the heterogeneous IoT deployments. This platform has been taken forward to implement a city scale infrastructure for IoT (SmartSantander project).

Another architectural proposal based on SoA in[14] proposes an integrated architecture incorporating various layers for application interface,

service management, device management, security and platform abstraction and devices layer. An SOA-based IoT architecture in an Enterprise Services Scenario is discussed in[24] which also shares a similar approach.

Semantic technologies are viewed today as a key technology to resolve the problems of interoperability and integration within heterogeneous world of ubiquitously interconnected objects and systems. Semantic technologies are claimed to be a qualitatively stronger approach to interoperability than contemporary standards-based approaches. Authors in[25] provide a good review of information modeling, ontology design, and processing of semantic data in the context IoT. According to authors most of the existing semantic tools and techniques have been created mainly for Web resources and have not taken into consideration the dynamicity of the physical environments and the constraints of the IoT resources. Lightweight and easy-to-use ontologies seem to have a better chance of being widely adopted and reused in order to create an interoperable platform across different domains and applications. They also suggest the application of linked data principles to the IoT domain to support creation of more interoperable and machine processable data and resource descriptions.

In[26,] the architectural proposal from UBIWARE incorporates a semantic oriented approach. In that interoperability is made possible using metadata and ontologies. In the proposed vision, each connected resource has an autonomous software agent and it is responsible for monitoring the state of the resource, making decisions, discovering the requests, and requesting external help when needed. An adapter or interface is used to connect the resource with its software agent. This adapter may include sensors and actuators, data structures and semantic adapter components as needed.

While there are several research groups contributing towards a unified architecture for IoT/M2M, notable outcomes from a standards perspective are from European Telecommunications Standards Institute (ETSI).[27] The main scope of the ETSI architecture is to specify a framework for developing M2M applications with a generic set of capabilities, independently of the underlying network. Its key architectural elements are domains, service capabilities, reference points, point of contact and resources. Beside these key architectural elements, the ETSI M2M architecture also defines procedures for security and bootstrap.

A service platform architecture that applied ETSI's network architecture concept[27] is presented in[28] with some modifications. They have implemented a prototype on "smart home automation" system based on this architecture which is divided into three domains, namely, the M2M service platform, M2M area network, and user/administrator domains. M2M service platform is developed using the connected objects operating system (COOS) open source middleware.[29] This seems to be a good proof of concept for the ETSI architecture.

## 4 Contributing Technologies for Internet-of-Things (IoT)

IoT technologies help applications to become intelligent through sensing of the physical world parameters and analysing the sensed data. Architecturally, in order to support extensive analytics of the sensed data in application deployment, cloud services over internet needs to be leveraged and it has to be done keeping security and privacy concerns in mind.

Keeping the above requirement in mind, we can classify some of the major contributing Technologies for IoT into four aspects—Connectivity, Security and Privacy, Analytics and Sensing Platforms. Connectivity needs to take care of transport of the sensed data to the cloud over internet. Security and Privacy needs to address the security concerns of the data transport along with associated privacy issues. Analytics is needed to impart intelligence to the Applications, very often in real-time. However the quality of the intelligence will also depend on the quality of the sensed data from the physical world and sensing platforms play a big role there.

In Table 1, we present these four technology aspects in details outlining the requirements, challenges and state-of-the-art of the technology.

## 5 IoT Software Platforms

We now take a look at some commercial and open source IoT software platforms for Data services, Device Management and Application Development in the market.

There are several mature device management servers available for consumer mobile devices such as phones and PDAs. Notable amongst these are the IBM Everyplace Device manager and the Microsoft System Center Mobile Device manager. For consumer premises network equipment such as home routers we have Broad Band Forum TR-069 compliant configuration servers like Axiros.[57] However, in the IoT case, because of lack of standardization, there is a wide variety of products in the market, each with their own proprietary offerings. Moreover, these offerings address only certain aspects of the entire gamut of IoT.

***Table 1.*** Major Contributing Technologies to IoT Services.

| Aspects | Requirements | Challenges | Technology |
|---|---|---|---|
| Connectivity | • Multiple protocol support<br>• Easy integration<br>• Sensor web enablement | • Constrained edge platforms<br>• High communication overheads (not suitable for M2M communications)<br>• Non-standard protocols and data semantics across verticals | *Low overhead protocols*<br>• Constrained Object Access Protocol (CoAP)[30]<br>• Message Queue Telemetry Transport (MQTT)[31]<br>*Standardization of Sensor data semantics*<br>• Open Geo-spatial Consortium—Sensor Web Enablement (OGC-SWE)[32] |
| Security and Privacy | • Authentication for Sensor Providers<br>• Access Control for Application Developers<br>• Data Encryption<br>• Data Privacy | • Key distribution for potentially trillions of devices<br>• Constrained edge platforms<br>• Balancing Privacy vs. Utility | *Low key distribution overhead*<br>• Identity Based Encryption (IBE)[33,34]<br>*Lightweight Systems*<br>• Light weight access control and authentication[35,36]<br>• Light weight encryption[37]<br>*Privacy Measurement*<br>• Privacy Quantification[38–40] |
| Analytics | • Scalable platforms<br>• Real-time analytics<br>• Domain modeling | • Reduce Computational Cost<br>• Inadequate knowledge representation for analytics<br>• Inadequate Cyber-physical system models | *Computing*<br>• Distributed Computing[41]<br>• Fog Computing[42]<br>*Analytics*<br>• Real-time in-stream processing and reasoning[43,44]<br>• Statistical modeling and System Identification[45,46] |
| Sensing Platforms | • Affordable<br>• Ubiquitous<br>• Unobtrusive<br>• Easy-to-deploy | • Constrained power in mobile phones<br>• Relatively new and yet to be mature areas for<br>a. Five senses computing<br>b. Cloud Robotics<br>c. UAV<br>• Trust measurement in Crowd-sensing<br>• Domain specific text analytics from Social Network sensing | • Mobile phone as a sensor (camera, microphone, accelerometer, magnetometer, gyroscope, GPS)[47]<br>• Five senses computing (3D vision, hearing, touch, smell, taste)[48]<br>• Cloud-controlled physically mobile sensing and actuating platforms (robots and unmanned aerial vehicles)[49–52]<br>• Crowd-sensing[53]<br>• Social Network as a soft sensor[54–56] |

One of the most well established software platforms for IoT/M2M is the Etherios Device Cloud.[58] It is a software suit that includes 'Device Manager' and 'Cloud Connector' for addressing device management and data services requirements of IoT systems. Etherios is offered as a cloud based service where service functions are offered as REST APIs. Clients connect to the Etherios cloud and access devices and device management functions using HTTP. In turn Etherios cloud uses a binary TCP protocol to connect to the Etherios Device Connect agent running on the device. The cloud based DM server also caches observations and events from the device. In addition, Etherios also lets the device log data in their databases for historical purposes. They have tools and APIs for facilitating application development. However details are not available on their standards compliance.

ILS Technology secureWISE connect[61] is a software platform that provides connectivity, control capabilities and some device management functionality. This platform, offered both as an on-premise installation as well as a hosted services, primarily focuses on creating a virtual, firewall friendly network connecting devices and enterprise datacenters in a single IP address space.

The SensiNode NanoService platform[62] consists of server side components and device library, and allows client applications to access resources on devices using HTTP REST interfaces. It uses 'Constrained Application Protocol' (CoAP), a protocol standard from IETF.[71] The server connects to the device using either CoAP, CoAP over SMS or HTTP. The server provides directory services for resources, a resource cache, search & lookup for resources and asynchronous communication of events to clients. The SensiNode NanoService in itself does not provide device management functions, but can be used as an efficient communication infrastructure over which device management services can be built.

No discussion on IoT platforms would be complete without Xively.[63] Xively, earlier known by other names such as Pachube and Cosm, is one of the earliest public IoT clouds, allows users to register, deploy and monitor devices on the cloud and associate sensor data feeds with these devices. It provides web service APIs for these services. Each registered user is provided with an API key which is used in every access to the platform. Xively provides libraries for use with selected device platforms for integrating with Xively APIs. Xively APIs can be accessed via transports such as HTTP, MQTT[64] and Sockets. Support for device management in Xively is however only partial. Xively provides a management console that provides limited services for device provisioning, activation and monitoring.

DeviceHive[65] is yet another cloud based device communication infrastructure that provides a common set of RESTful web service APIs for access from both web clients as well as devices. DeviceHive does not provide management functions on its own, rather it provides a framework for sending arbitrary commands to devices and receive notifications in response. Using this framework, additional functionality can be built.

52North Sensor Web[66] is an open Source project that is built to implement the Open Geospatial Consortium (OGC) Sensor Web Enablement interface standards.[67] A key sub-project is the Sensor Observation Service platform that provides access to sensor data encoded in SensorML and capture and query of observations based on the OGC Observation & Measurement schema and interface specifications. The Sensor Observation Service provides web service APIs for sensor registration, inserting observations and making queries. Virtually any kind of sensor and measurement procedure can be supported. This platform allows spatio-temporal queries on stored sensor observation data.

Axeda,[59] part of the AT&T M2M service delivery platform, provides fairly extensive support for asset tracking and monitoring applications. It provides a rich data model and data services for capturing, storage and query of devices, assets, organization structures, users, locations and alarm data. It also provides a rules engine for rule based processing of incoming data feeds from devices. Rules may be processed on the edge devices as well as on centralized servers. It allows users to run customized scripts to extend the platform provided features. Axeda provides web service APIs for integration to the platform and also supports integration to commercial ERP systems via an integration framework. Axeda is also embedded into the AT&T M2M Control Center solution.[60]

SensorLogic[68] is an IoT platform with elements of service management, device management as well as application & data management. SensorLogic provides device connectors for connecting devices to the platform over SMS and TCP protocols. It provides operator provisioning, device activation and status monitoring services as well. However, unlike other known platform, it has strong support for business rules creation and complex event processing. It has a message routing facility that intelligently distributes the incoming sensor feed to destination services running on the platform. Platform services include message storage, geo-coding services, geo-fencing, complex event processing, alarms and notification services. Platform services are made available to users via HTTP/REST APIs and JavaScript libraries.

ThingWorx[69] is an application development platform that provides application developers with tools for model driven development of IoT applications. It provides a data and ontological model for storage and semantic query/search of device data. The data storage engine, based on graph databases, is able to store device data, device connections and relationships. ThingWorx supports search, query and analysis on the data storage engine. It also provides a mashup builder that can be used to compose sophisticated applications.

Qualcomm Life 2Net platform[70] is wireless health platform that enables secure collection of biometric and health data from devices to client applications and databases. The 2 Net platform provides gateway devices and cloud based backend systems for data management, device management, application integration and connectivity service provisioning and management. Thus it covers all aspects of IoT platform requirements, though it is designed for a specific application domain. The platform is certified to work on major wireless operator networks. In addition to

the platform, Qualcomm Life also provides 'Network Operations Centers' services worldwide for 24×7 monitoring and management of connected health devices. The platform also meets regulatory standards like HIPAA.

While there are quite a few platform solutions their performances are yet to be proved on large scale applications. Many of them are deployed in some pilot level or small scale applications. The proprietary standards followed by many of the platforms might have to give way to the upcoming standards from reputed bodies such as ETSI. One M2M is a consortium[72] attempting to harmonize IoT/M2M standards from a worldwide perspective.

## 6 Conclusions

In this paper we put forth key requirements of a software platform for IoT/M2M services and surveyed key developments from related areas. It is observed that there is a trend towards architecting horizontally integrated platforms addressing multi-domain services, moving away from vertical specific silos. Even in the case of domain specific applications, the platform architecture is desired to be horizontal and capable of supporting multiple domains. Interoperation of applications and devices is another aspect of improvement. Beyond a standardized interfaces approach for interoperability, the potential of using semantic oriented architectures is yet to be realized in a substantial way. From the overall platform architecture perspective it is important to address the requirements of all the stake holders including application developers. The importance of developers' requirements need to be addressed by providing efficient and user friendly development environment along with the software platform. This is still a shortcoming with many of the platforms reviewed in this paper. Analytics as a service for faster development of applications is another gap to be addressed. This would enable the delivery of analytics intensive applications through efficient service composition. Service providers require features to enable accounting and billing for the services delivered and a mature approach for this need to be integrated to the software platform. Future platforms need to consider emerging standards from ETSI and One M2M.

Received 14 July 2013.

## References


1. Jayavardhana Gubbia, Rajkumar Buyyab, Slaven Marusic, Marimuthu Palaniswami, Internet of Things (IoT): A vision, architectural elements, and future directions, Elsevier journal on Future Generation Computer Systems 29 (2013) 1645–1660.
2. INFSO D.4 Networked Enterprise & RFID INFSO G.2 Micro & Nanosystems, in: Co-operation with the Working Group RFID of the ETP EPOSS, Internet of Things in 2020, Roadmap for the Future, Version 1.1, 27 May 2008.
3. I. Toma, E. Simperl, Graham Hench, A joint roadmap for semantic technologies and the internet of things, in: Proceedings of the Third STI Roadmapping Workshop, Crete, Greece, June 2009.
4. A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin, V. Terziyan, Smart semantic middleware for the internet of things, in: Proceedings of the Fifth International Conference on Informatics in Control, Automation and Robotics, Funchal, Madeira, Portugal, May 2008.
5. W. Wahlster, Web 3.0: Semantic Technologies for the Internet of Services and of Things, Lecture at the 2008 Dresden Future Forum, 1708 June 2008.
6. Soma Bandyopadhyay, Munmun Sengupta, Souvik Maiti and Subhajit Dutta, "Role Of Middleware for Internet Of Things: A Study", International Journal of Computer Science & Engineering Survey (IJCSES) Vol. 2, No. 3, August 2011.
7. S. Gilani, The promise of M2M: how pervasive connected machines are fueling the next wireless revolution, in Embedded Systems White Paper, 2009. http://www.mentor.com
8. K. Kouji, N. Yoshitaro, S. Tadashi, M2M service platform to support carrier cloud. NEC Tech. J. 5(2), 116–121 (2010).
9. Dominique Guinard, Iulia Ion, and Simon Mayer, In Search of an Internet of Things Service Architecture: REST or WS-*? A Developers' Perspective.
10. N.B. Priyantha, Aman Kansal, Michel Goraczko, and Feng Zhao. Tiny web services: design and implementation of interoperable and evolvable sensor networks. In Proc. of the 6th ACM conference on Embedded Network Sensor Systems (SenSys'08), pages 253–266, Raleigh, NC, USA, 2008. ACM.
11. Luigi. Atzori et al., The Internet of Things: A survey, Elsevier Journal of Computer Networks (2010), doi:10.1016 j.comnet.2010.05.010.
12. De Deugd, R. Carroll, K. Kelly, B. Millett, J. Ricker, SODA: service oriented device architecture, IEEE Pervasive Computing 5 (3) (2006) 1732, 94–96.
13. J. Pasley, How BPEL and SOA are changing web services development, IEEE Internet Computing 9(3) (2005) 60–67.
14. P. Spiess, S. Karnouskos, D. Guinard, D. Savio, O. Baecker, L. Souza, V. 1736 Trifa, "SOA-based integration of the internet of things in enterprise services," in: Proceedings of IEEE ICWS 2009, Los Angeles, Ca, USA, July 2009.
15. Hydra Middleware Project, FP6 European Project, http:// www.hydramiddleware.eu
16. R. Fielding. Architectural styles and the design of network-based software architectures. Phd thesis, 2000.

17. Cesare Pautasso, Olaf Zimmermann, and Frank Leymann. Restful web services vs. big web services: making the right architectural decision. In Proc. of the 17th international conference on World Wide Web (WWW '08), pages 805–814, New York, NY, USA, 2008. ACM.
18. Cesare Pautasso and Erik Wilde. Why is the web loosely coupled?: a multi-faceted metric for service design. In Proc. of the 18th international conference on World Wide Web (WWW '09), pages 911–920, Madrid, Spain, April 2009. ACM.
19. Leonard Richardson and Sam Ruby. RESTful web services. O'Reilly Media, May 2007.
20. Dominique Guinard, Vlad Trifa, and Erik Wilde. A Resource Oriented Architecture for the Web of Things. In Proc. of the 2nd International Conference on the Internet of Things (IoT 2010), LNCS, Tokyo, Japan, November 2010. Springer Berlin Heidelberg.
21. Thiago Teixeira, Sara Hachem, Val'erie Issarny, and Nikolaos Georgantas, Service Oriented Middleware for the Internet of Things: A Perspective.
22. Jesus Bernat Vercher1, José M. Hernández-Muñoz1, Luis A. Hernandez Gomez, Alfonso Tierno Sepulveda, "An Experimental Platform for large scale research facing FI-IoT scenarios",http://www.smartsantander.eu/downloads/Presentations/usn_tid_futurenetworksummit_v6.pdf
23. Bernat, J., Marín, S., González. A., Sorribas, R., Villarrubia, L., Campoy, L., Hernández, L. Ubiquitous Sensor Networks in IMS: an Ambient Intelligence Telco Platform. ICT Mobile Summit, 2008. 10–12 June, Stockholm.
24. Patrik Spiess, Stamatis Karnouskos, Dominique Guinard, Domnic Savio, Oliver Baecker; Luciana Moreira S´a de Souza, and Vlad Trifa; "SOA-based Integration of the Internet of Things in Enterprise Services".
25. Payam Barnaghi, Wei Wang, Cory Henson and Kerry Taylo, "Semantics for the Internet of Things: early progress and back to the future".
26. A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin, and V.Y. Terziyan, "Smart Semantic Middleware for the Internet of Things", in Proc. ICINCO-ICSO, 2008, pp. 169–178.
27. ETSI TS 102 690 v2.0.6, ETSI technical specification, machine-to-machine communications (M2M); functional architecture. (2012). http://portal.etsi.org
28. Eui-Jik Kim1 and Sungkwan Youm, Machine-to-machine platform architecture for horizontal service integration, EURASIP Journal on Wireless Communications and Networking 2013,
29. J. Audestad, I. Gronbak, S. Svaet, Connected objects platform specification version 1, R&I Research Report, 1–66. (2009).
30. Shelby, Zach, Klaus Hartke, and Carsten Bormann. "Constrained application protocol (coap)." (2013). http://tools.ietf.org/html/ietf-core-coap-14.txt
31. Stanford-Clark, A.S., and Hong Linh Truong. "MQTT for sensor networks (MQTT-S) protocol specification." (2008). http://mqtt.org/MQTT-S_spec_v1.1.pdf
32. Botts, Mike, et al. "OGC® sensor web enablement: Overview and high level architecture." GeoSensor networks. Springer Berlin Heidelberg, 2008. 175–190.
33. Boneh, Dan, and Matthew Franklin. "Identity-based encryption from the Weil pairing." SIAM Journal on Computing 32.3 (2003): 586–615.
34. Baek, Joonsang, et al. "A survey of identity-based cryptography." Proc. of Australian Unix Users Group Annual Conference. 2004.
35. Johnson, Henric, "Toward Adjustable Lightweight Authentication for Network Access Control", Elektronisk resurs.—2005.—ISBN: 91-7295077-3.
36. Vajda, István, and Levente Buttyán. "Lightweight authentication protocols for low-cost RFID tags." Second Workshop on Security in Ubiquitous Computing–Ubicomp 2003. 2003.
37. Zhen, Jianliang, et al. "A lightweight encryption and authentication scheme for wireless sensor networks." International Journal of Security and Networks 1.3 (2006): 138–146.
38. Aggarwal, Charu C., and S. Yu Philip. A general survey of privacy-preserving data mining models and algorithms. Springer US, 2008.
39. Bertino, Elisa, Dan Lin, and Wei Jiang. "A survey of quantification of privacy preserving data mining algorithms." Privacy-preserving data mining. Springer US, 2008. 183–205.
40. Chellappa, Ramnath K., and Raymond G. Sin. "Personalization versus privacy: An empirical examination of the online consumer's dilemma." Information Technology and Management 6.2–3 (2005): 181–202.
41. P. Ghosh, K. Basu, and S.K. Das, "Cost-Optimal Job Allocation Schemes for Bandwidth-Constrained Distributed Computing Systems," in HiPC, 2005.
42. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in Proceedings of the first edition of the MCC workshop on Mobile cloud computing, New York, NY, USA, 2012.
43. Della Valle, Emanuele, et al. "A first step towards stream reasoning." Future Internet—FIS 2008. Springer Berlin Heidelberg, 2009. 72–81.
44. Della Valle, Emanuele, et al. "It's a Streaming World! Reasoning upon Rapidly Changing Information." Intelligent Systems, IEEE 24.6 (2009): 83–89.
45. Rissanen, Jorma. Information and complexity in statistical modeling. Springer Publishing Company, Incorporated, 2007.
46. Söderström, Torsten, and Petre Stoica. System identification. Prentice-Hall, Inc., 1988.
47. Lane, Nicholas D., et al. "A survey of mobile phone sensing." Communications Magazine, IEEE 48.9 (2010): 140–150.
48. "IBM reveals five innovations that will change our lives within five years", http://phys.org/news/2012–12-ibm-reveals-years.html#jCp
49. Guizzo, Erico. "Robots with their heads in the clouds." Spectrum, IEEE 48.3 (2011): 16–18.

50. Hu, Guoqiang, Wee Peng Tay, and Yonggang Wen. "Cloud robotics: architecture, challenges and applications." Network, IEEE 26.3 (2012): 21–28.
51. Stuart M. Adams and Carol J. Friedland, "A survey of unmanned aerial vehicle (UAV) usage for imagery collection in disaster research and management", 9th International Workshop on Remote Sensing for Disaster Response, 2011.
52. Therese Skrzypietz, "Unmanned Aircraft Systems for Civilian Missions." Policy Paper by Brandenburg Institute for Society and Security, 2012.
53. Ganti, Raghu K., Fan Ye, and Hui Lei. "Mobile crowdsensing: Current state and future challenges." Communications Magazine, IEEE 49.11 (2011): 32–39.
54. Samuel Alan Stewart, Syed Sibte Raza Abidi, "Using Social Netowrk Analysis to understand web 2.0 Communities", Medicine 2.0 2011, Stanford, USA.
55. Maynard, Diana, Bontcheva, Kalina and Rout, Dominic., "Challenges in developing opinion mining tools for social media", Proceedings of@ NLP can u tag\# user_generated_content, 2012.
56. Plantie, M., Crampes, M., "Survey on Community Detection", Social Media Retrieval, Computer Communications and Networks, Springer-Verlag, London, 2013.
57. Axiros—http://axiros.com/offerings/competenceintr-069-andbeyond/axessacs-tr-069-autoconfiguration-server.html
58. Etherios Device Cloud—http://www.etherios.com/products/devicecloud/
59. Axeda M2M Cloud Service—http://www.axeda.com/products
60. AT&T M2M 360 Control Center—http://www.business.att.com/enterprise/Family/mobility-services/machine-to-machine/
61. ILS Technology secureWISE connect—http://www.ilstechnology.com/securewise-connect
62. SensiNode NanoService Platform—http://www.sensinode.com/EN/products/nanoservice.html
63. Xively—https://xively.com/
64. MQTT—Message Queue Telemetry Transport—http://mqtt.org
65. Device Hive—http://www.devicehive.com/documentation
66. 52North Sensor Web—http://52north.org/communities/sensorweb/
67. Open Geospatial Consortium Sensor Web Enablement—http://www.opengeospatial.org/domain/swe
68. SensorLogic Application Enablement Platform—http://m2m.gemalto.com/application-enablement.html
69. ThingWorx—http://www.thingworx.com/
70. Qualcomm Life 2Net Platform—http://qualcommlife.com/wireless-health
71. Constrained Application Protocol (CoAP)—IETF Draft Standards, March 2013—http://tools.ietf.org/html/draft-ietf-core-coap-14
72. oneM2M—http://www.onem2m.org

**Balamuralidhar P.** is a Principal Scientist and Head of TCS Innovation Lab at Tata Consultancy Services Ltd (TCS), Bangalore. He has obtained Bachelor of Technology from Kerala University and Master of Technology (MTech) from IIT Kanpur. His Ph.D. is from Aalborg University, Denmark in the area of Cognitive Wireless Networks. Major areas of current research include different aspects of Cyber Physical Systems, Sensor Informatics and Networked Embedded Systems. Before TCS his research careers were with Society for Applied Microwave Electronics Engineering & Research (SAMEER) Mumbai and Sasken Communications Ltd Bangalore.

He has over 25 years of research and development experience in Signal Processing, Embedded Systems and Wireless Communications. He has over 60 publications in various international journals and conferences and over 20 patent applications. Balamuralidhar was the leading TCS participation in two EU FP6 research consortium projects namely My Adaptive Global NET (MAGNET) and End to End Reconfigurability (E2R) in the area of next generation wireless communications. He is also contributing to TCS participation in National bodies like Broadband Wireless Consortium India (BWCI), Global ICT Standards for India (GISFI). In GISFI he is chairing the Internet of Things Workgroup.

**Prateep Misra** is a Research Area Manager in TCS Innovation Labs and is leading the development of platforms for Internet-of-Things applications. He has over 20 years experience in the IT Industry in areas such a software development, research, technology consulting and software quality assurance. Current focus areas include IT Infrastructure architecture & design, IT transformation, storage systems, large scale & real-time analytics platforms and cloud computing. As a head of RFID Center-of-Excellence in TCS, he was responsible for over 25 projects executed globally in the area of RFID. Prateep was also responsible for design of IT infrastructure for a major SaaS offering from TCS. He holds a B. Tech in Instrumentation Engineering and M.Tech in Electrical Engineering, both from Indian Institute of Technology, Kharagpur.

**Arpan Pal** is a Ph.D. from Aalborg University Denmark and is a senior member of IEEE. He had received his B.Tech and M.Tech from Indian Institute of Technology, Kharagpur, India in Electronics and Telecommunications.

He has more than 20 years of experience in the area of Signal Processing, Communication and Real-time Embedded Systems. Currently he is with Tata Consultancy Services (TCS), where he is heading research at Innovation Lab, Kolkata. He is also a member of Systems Research Council of TCS. His main responsibility is in conceptualizing and guiding R&D in the area of cyber-physical systems and ubiquitous computing with focus on applying the R&D outcome in the area Intelligent Infrastructure.

His current research interests include Mobile phone and Camera based Sensing and Analytics, Physiological Sensing, M2M communications and Internet-of-Things based Applications with focus on Energy, Healthcare and Transportation verticals. He has more than 40 publications till date in reputed Journals and Conferences along with a couple of Book Chapters. He has also filed for more than 35 patents and has 5 patents granted to him. He is an editor for IEEE Transactions on Emerging Topics in Computing for the special issue on Emerging Computing Technologies for Resilient & Robust Intelligent Infrastructure.

He had been earlier with Defense Research and Development Organization (DRDO) of Indian Govt. working on Missile Seeker Signal Processing. He has also worked with Macmet Interactive Technologies, leading their real-time systems group in the area of Interactive TV and Set-top boxes.