A novel versatile architecture for Internet of Things

Luka Milić[*] and Leonardo Jelenković[**]

[*] University of Zagreb, Faculty of Organization and Informatics, Varaždin, Croatia
[**] University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb, Croatia
luka.milic@foi.hr, leonardo.jelenkovic@fer.hr

Abstract – This paper presents an overview of contemporary architectures for Internet of Things and then introduces a simple novel architecture. The main goal of proposed architecture is to remain simple but still applicable to any Internet of Things environment. Specific applications of IoT should be implemented in application layer, using proposed architecture as backbone. Proposed architecture isn't yet fully defined, but ideas on which is based are well defined and should provide straightforward design and implementation.

I. INTRODUCTION

Internet of Things (IoT) is a relatively new paradigm of connecting anyone, anything, anywhere and anytime. Despite being relatively fresh, the idea is much older. Even though the term IoT is first mentioned by Kevin Ashton in 1999, similar networks were discussed through the 90s. Also, IoT does not necessarily include new technologies, but often only a new use of the old or current technologies, like Radio Frequency Identification (RFID) and Wireless Sensor Networks (WSN).

However, only recently, in the current decade or so, IoT has been studied more intensively, as the technology has been rapidly evolving to enable the IoT. IoT has gone beyond Ashton's original idea of connecting *things* in supply chains to the *Internet* and grew to occupy almost every part of human lives. It has been successfully applied to, for example, climate monitoring, transport and road safety, home automation and building monitoring, health care, supply chain, agriculture and rural development, border security and military application, etc.

IoT is commonly defined as a network of many resource-constrained nodes connected to local networks and then, through the higher layers of IoT platform, connected to everyone and everything. A typical IoT architecture looks as illustrated by Fig. 1. IoT is commonly divided into three layers, possibly with two sub-layers each, as seen in the picture.

The lowest layer, encompassing sensors and the physical world, is called the perception layer. The most prominent technologies in the perception layer are RFID for identifying and sensing things, WSN for sensor communication, and lightweight wireless communication protocols for WSNs and other types of networks. ZigBee seems to be the current state of the art protocol for such cases. Bluetooth is another protocol, but not as adapted for resource-constrained devices. Recently, there has been a lot of effort put into Internet Protocol version 6 Low-Power Personal Area Networks (6LoWPAN) and Constrained Application Protocol (CoAP) for communicating on higher layers. For things less constrained by power or performance, sometimes even their full counterparts Internet Protocol version 6 (IPv6) and Hypertext Transfer Protocol (HTTP) are used.

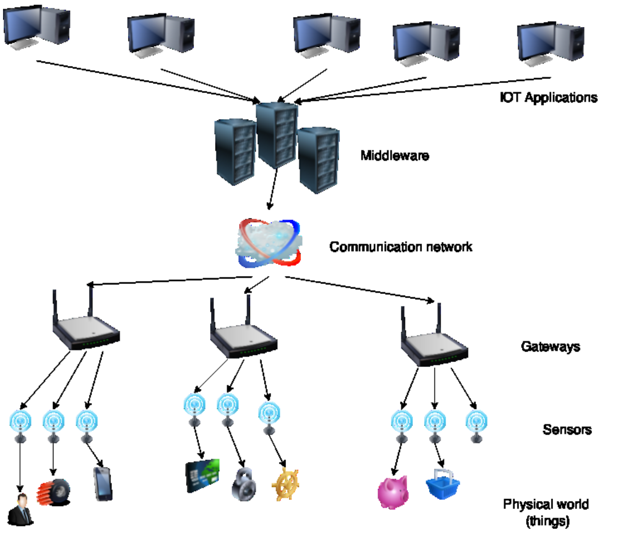

Figure 1. A typical IoT architecture

On the network layers, consisting of gateways communicating through the broadband network, there have been suggestions to use mobile networks, the Internet or even completely new networks. However, recently the use of the current Internet has been taken for granted.

On the application layer, here consisting out of IoT middleware and user applications, the research is extremely diverse. There are many unique or less unique approaches proposed by researchers in order to connect presumably at least billions of things in a working network.

Still, a significant amount of research that has been put into IoT neither means the field has been sufficiently explored nor that satisfying global solutions have been found. A common person still does not know what IoT is, and the deployment of IoT is only in nascent stages, with the little that has been deployed varying in technology and coverage. Also, the researchers have mostly been concentrated on building IoT for specific cases, effectively building small isolated networks of things, not Internet of Things but rather Intranets of Things. We didn't find adequate architecture proposals in rest of the research, architecture that could be used as base for all IoT applications. Therefore we designed one architecture that could be used as base for implementing IoT.

The rest of the article is organized as follows. In Section II the state of the art is presented and similar conclusions drawn. In Section III our new architecture for IoT is proposed. In Section IV the conclusions and future work are laid out.

II. Related Work

We will now try to summarize main research directions in IoT architectures. Usually, first ones referenced are architectures backed up by global groups. Some of these architectures will be presented here, but it should be noted that there is not much scientific research based on those architectures and we do not think that they have necessary features for a global IoT. Two ongoing Seventh Framework Programme (FP7) projects will also be presented, IoT-A and OpenIoT, as they are European Union (EU) projects relevant to the local scientific community, but also very large projects with important investors.

First architecture is the EPCglobal project [1]. It is promulgated by the EPCglobal organization, successor of the Automatic Identification (Auto-ID) Labs where the term IoT was coined. The GS1 architecture offers three sets of standards. Those are standards for creating a variety of codes for things, standards for representing these codes by graphics and RFIDs, and standards for manipulating the data captured by reading these codes. It is more concerned with creating business models like supply chains than creating a global network.

The second architecture is the Machine Type Communication (MTC) [2] by Third Generation Partnership Project (3GPP). MTC is a set of standards for communication between different things and the core network. It is focused on implementing the communication through Long Term Evolution (LTE) networks and mobile phones. No upper architecture seems to be emerging. Paper [3] presents a potential one, putting servers and applications over it. Communication is rather complex, requiring a lot of telecommunication knowledge in order to push messages from one side to another.

The third architecture is the Machine to Machine (M2M) [4] done by European Telecommunication Standards (ETSI). M2M aims to create applications, built upon service capabilities, in both the upper layer and the gateways, but also the things, which do not need a gateway in that case. It does so by defining three sets of interfaces. These are M2M application interface, device application interface, and M2M to device interface. There does not seem to be concrete technologies defined.

The fourth chosen commonly referenced architecture is the Near Field Communication (NFC) [5] put forward by the NFC Forum, mainly Nokia. This system is used only for identifying and capturing things. It is already built into many smartphones. It does not seem to aspire to building higher layers, but only uses RFID to communicate with sensors and let them interpret the data.

The most successful of these global groups seem to be the Internet Engineering Task Force (IETF). They have constructed two protocols, the more successful of which is the Constrained Application Protocol (CoAP). Recently, it has generated many research papers and its architectures will be presented a bit later. CoAP alone is not complete architecture. It is a counterpart for the Hypertext Transfer Protocol (HTTP) for resource-constrained devices.

The second protocol is the Extensible Messaging and Presence Protocol (XMPP), an even higher-layer protocol family for near real-time communication. An example system in [6] builds a basic platform for bidirectional communication based on publish-subscribe and request-response.

Currently, various visions of IoT seem to converge in implementing IoT using the current Internet. There are architectures like MobilityFirst [7], which envision a complete overhaul of the current network. The researcher developed a new vision or the internet, with the main goal being extreme mobility of the members, which they see as the ideal environment for the IoT. In the paper they specify exactly how IoT is combined with the network. Still, visions of replacing the current internet seem more and more far-fetched.

Searching for more implementable global architectures, the first thing visible is the huge work targeted on creating small, isolated IoTs, here dubbed "Intranets of Things". After a brief search on, for example, Institute of Electrical and Electronics Engineers (IEEE) Explore, many narrow niches like electronic commerce, smart supermarkets, rural patient tracking smart gas tanks, smart scenic sites, smart food, electronic identities and much more pop out. A good architecture needs to be flexible, meaning it can be used for many use cases, unlike those. Only architectures intended for a global IoT will be further analyzed. Also, there has been much work in order to completely fixate the IoT to the current Internet. Host Identity Protocol [8] has been used for IPv4 and IPv6 Low-Power Personal Area Networks (6LoWPAN) for IPv6. The paper referenced deals with address translations.

Some common architecture proposals will now be presented. The first is Service Oriented Architecture (SOA). An example in [9] outlines a network of service entities. These are connected with services, actually general data, being sent through channels or links. It is just a sketch of a potential model and its behavior.

SOA is borrowed from the Semantic Web, and so is Representational State Transfer (REST). Actinium [10] is an example of a RESTful runtime container for scriptable IoT applications. It uses CoAP for communicating in a real network. The user needs to implement applications in a RESTful fashion to communicate via CoAP, and still seems to only get an Intranet of Things. As already said, CoAP is a HTTP counterpart. But even if HTTP is great, binding to four layers of protocols does not have to be.

Another Semantic Web technology is the Simple Object Access Protocol (SOAP). Paper [11] defines a things management protocol based on it. It relies on Web Services Description Language (WSDL) to define three types of operations, getting information, getting next information and setting information. It does not provide a higher layer.

The use of the ontologies is presented in [12] in order to refine the crude data received from things. Having defined a data model ontology in Web Ontology Language (OWL) and described data sources, data providers, and the data, and also defining concrete ontologies, an algorithm is provided in order to integrate them.

Another CoAP proposal is [13]. It is focused on the EPC and creates a network using ZigBee on the lower layer and CoAP on the higher, also utilizing Universal Plug and Play (UPnP) for discovery mechanisms.

Paper [14] implements SOA through CoAP with the help of peer-to-peer (P2P). A new P2P protocol is put forward in order to create such networks, used for their Distributed Hash Tables (DHT). All in all it seems rather complex, and we believe that if the Internet of the 80's started with a complex architecture, there would be no Internet of today.

There is also a notion of message-oriented architectures. Paper [15] discusses a message-oriented middleware as a set of devices, which can receive, send, store and forward messages between devices, acting as a temporary database with different channels acting like dynamically created pipes.

Having mentioned P2P, reference [16] presents a system built especially upon it. It uses the DHT for identifying and locating things, and also uses other nodes for publishing services. It is still rather abstract, since it is just a helper network.

P2P is just another way of dealing with scalability. Special systems with a focus just on the scalability have emerged.

Some work is focused on treating IoT as a database. This system [17] creates three storage layers as an IoT abstraction and uses database indexes to navigate it. The idea is expanded with more indexing types and supporting infrastructure.

A common buzzword in the scalability is the self-organization. For example, architecture [18] is based on the nature-inspired algorithm of endocrine systems. It uses so-called hormones to, not interfering with each other, communicate between things and slightly modify their behavior. It does not consider that the things will be joined in local networks.

Another architecture is [19], which presents a middleware platform built upon immobile WSNs, which and then successfully spatially grouped and included to the global network.

Sometimes, there is talk about autonomic architectures, which is not connected with self-organization, but seems to emphasize things cannot work with IP. Paper [20] outlines an architecture based on multiple planes with their agents and protocols to abstract the IP in the communication.

Supposed to complex networks, there are systems for only helping existing networks. For example, Virtual Sensor Editor [21] integrates nodes from different networks using graphical component programming.

There are also security architectures built that way, on top of the networks, which seems to introduce more overhead and effort that just including security.

For example, [22] presents a security system based on the HIP Diet Exchange (DEX), extending the key management protocol by a number of ideas like giving the attacker increasingly difficult puzzles. It is also expanded to other key management protocols, but nothing else.

We believe that, conversely, security should be one of the basic features of the IoT.

Also, simplicity is another requirement. Only when our homes and offices get IoT people will care more about it, and that will not be possible with a complicated network.

Finally, some complete architecture proposals will be presented. Paper [23] shows a system with extremely thin client nodes, where the servers are used for presenting their data. It also presents compressed TCP and compressed HTTP, which are used for communication. No security seems to be emphasized, even though things need to implement four layers.

Paper [24] seems to be a large project for a complete architecture with proprietary physical protocols and IPv6, but it is still a work in progress. Gateway servers are envisioned to be used to connect things to the Internet.

Paper [25] sees IoT as a network of gateways without intermediate servers. Not too much is detailed, but it appears to be intended for Intranets of Things. Still, it is modeled to be modular and scalable.

Paper [26] shows a system based on a SOA spin-off called Event-Condition-Action (ECA). Even though envisioning a complete architecture, only the platform intended for making triggers upon the ECA rule and communication with neighboring layers are detailed.

Reference [27] utilizes a similar system like ours, based on M2M and using HTTP with REST on upper layers. It also enables discovery, connections with non-smart things, associating metadata and actuator control.

OpenIoT project [28] develops an open source middleware for getting information from sensor clouds, without having to worry about what exact sensors are used. The OpenIoT architecture is comprised by seven main elements that belong to three different logical planes. In the Utility-App Plane – the application layer, there are three modules: Request Definition – for users to define requests to the IoT, Config & Monitor – for administrators to configure and monitor the middleware and things, and Request Presentation – for presenting responses from the IoT. In the Virtualized Plane – the service layer, there are three modules too: Scheduler – for handling requests to the IoT, Cloud Data Storage – for a cloud database of the IoT, and Service Delivery & Utility Manager – for delivering responses from the cloud and tracking metrics of the data. In the Physical Plane – the physical layer, there is only one module: Sensor Middleware – for communicating with the things.

IoT-A [29] creates an architectural reference model for the IoT. Four things are identified as building blocks of the model. The vision summarizes the rationale for providing an architectural reference model for the IoT. Business scenarios & stakeholders are the drivers of the architecture work. The IoT Reference model provides the highest abstraction level for the definition of the IoT-A Architectural Reference Model. The IoT Reference Architecture is the reference for building compliant IoT architectures. The Reference Model mainly defines five things as abstractions. The entity model defines Physical Entities (roughly Devices), Virtual Entities (roughly Services), and Users. The data model defines information types. The functional model defines layers, lower to higher, as Device, Communication, IoT Service, Virtual Entity, IoT Business Process Management, and Application. The communication model defines communication layers, higher to lower, as Data, End to End, ID, Network, Link, and Physical. The security model defines security features and their layering. The Reference Architecture mainly defines functionalities each mentioned actor has to implement. An extensive list of desired functionalities is given. Also, there are UML Use Case Diagrams, UML Sequence Diagrams, and interfaces, defined.

III. Overview od proposed architecture

Proposed architecture could be roughly divided into four layers:

1. Things layer
2. Communication layer
3. Data storage and control layer
4. Application layer.

Similar division is presented on most IoT architectures. However, operations on those layers and communication between them differentiates architectures. Fig. 2 presents an overview of presented architecture. Short description of concepts for each layer follows.

Things in things layer have their own operation mode we don't model. We assume they need to communicate with layer 3 to fulfill their purpose. Since things could be various, from very simple battery operated devices to powerful computers, we don't define multi-layer communication stack – communication mechanism is broadcast on physical layer. Other things might catch that broadcast message and broadcast it further. We don't propose special communication protocol on physical layer, any current or future will work. We just need them to carry a message with IoT data (described later).

Communication layer provide connectivity between things and data storage and control layer (layer 3). We assume that gateways could be used in this layer to catch message that things broadcast and forward it (using TCP/IP or some application protocol) to layer 3. Gateways will also forward messages in other direction: from layer 3 to devices.

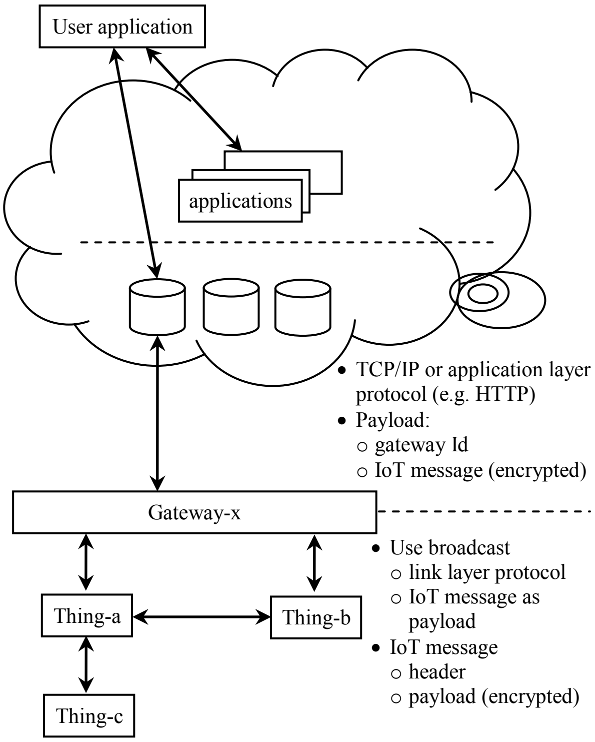

Figure 2. Overview of proposed architecture

Data storage and control layer contains databases with information about devices (and how to reach them - through which gateways), users owning devices, past exchanged information with devices (both received from devices and send to devices), and triggers for control. Triggers could be used to start some action when an event occurs. Events could be simply related with sending and receiving data from one thing or a complex trigger that involves evaluation of received data, combination of data in databases, periodic events (periodically request update from things) and similar. Action on triggers could vary, from simple data retrieval request from things to a start of complex operation implemented in application layer.

Application layer is where things with their data and operations are used for creating some benefit for a end user, such as sensor readings, command dispatch to remote devices and similar. We don't design this layer, we perceive that most required operations in this level can be created using described lower layers we define.

Things with Internet connectivity could act as gateways (directly communicate with layer 3) or connect with gateway using TCP/IP. There is no need to use broadcast if a thing is smarter and can detect internet connectivity, or connectivity with gateway. For connection to gateway it doesn't have to use full TCP/P stack – physical layer protocol could be used, just without broadcast.

Using broadcast in layer 1 looks terribly ineffective, choking the network with packets. However, there are a number of facts helping here. Firstly, the packets are supposed to be short and rare. Secondly, constant ping-pong of the packets must be mitigated using a Time to Live (TTL) mechanism, analogous to the one used by the Internet Protocol (IP). Thirdly, each packet has a local-network identifier. A packet that ends up in a different local network could simply be dropped. Fourthly, if a thing receives a packet meant for it, it will naturally not forward the packet. Fifthly, as every packet has its identifier so that already forwarded packet can also be dropped.

Some gateways can have a subset of layer 3 functionality, enabling operating with things from local network only, in which applications could use local gateway instead of layer 3 from cloud. Such gateways could still communicate with full layer 3 implementation in cloud (and forward a copy of messages to them).

IV. IoT MESSAGE FORMAT AND SECURITY ISSUES

IoT messages must contain relevant information - payload (per thing specific) but must also contain communication data - header. Header should contain:

a) protocol identifier

b) time-to-live (TTL)

c) thing's identifier (e.g. physical layer address)

d) packet identifier

e) network identifier

f) payload length.

Identifiers could be used to manage messages. A thing can process messages sent to it, broadcast all others or just ones that belong to same network. Packet identifier could be use to manage packet flow (e.g. discard duplicate packets).

Payload could be in various formats but for maximal extensibility we propose simple string, carrying a message (that could be in e.g. Java Simple Object Notation - JSON) with values.

Payload must be encrypted with thing's password (known to thing and layer 3 only). Advanced Encryption System (AES) or similar algorithm could be used to provide sufficient security on this layer.

Gateways could communicate with things on layer 1 with previously described principles. For communication with layer 3 in cloud, standard TCP/IP or some application layer could be used (like HTTP). In a simple implementation, the UDP protocol could be used, where an UDP packet will contain IoT message forwarded from layer 1 or for layer 1. This message could be encrypted with gateway's password (known to layer 3 and gateway only) which could solve security issues on this level.

Security of communication protocols between applications (layer 4) and layer 3 is out of scope of this proposal. For example, some application layer protocol could be used (e.g. SSL).

Management in layer 3 must ensure that only authenticated users (their applications) could manipulate with data and devices that belong to them or they are given appropriate privileges by owners or administrators.

V. COMPARISON WITH IoT-A AND OPENIoT

In this section we compare presented architecture with architectures from projects IoT-a and OpenIoT, as most recent and probably most refined architectures for IoT today.

Comparing proposed architecture with the OpenIoT, the biggest difference is that proposed architecture is incomparably simpler. Proposed architectures is concerned about protocols for communicating with things, protocols for communicating with servers in cloud, definitions of gateways and definitions of a cloud. Also, the Sensor Middleware in the OpenIoT relies heavily on running their web server X-GSN written in Java. In proposed architecture the local network with arbitrary link protocols are used, inside of which is encapsulated our own simple protocol, while gateways speak with the cloud over an arbitrary network layer.

Comparing the architecture with the IoT-A, it is seen that IoT-A defines no physical architecture. Gateways, for example, are only briefly mentioned saying they can possibly be put in the middle to help with the communication. We, on the other hand, put forward actual physical architecture and actual communication. IoT-A only describes what needs to exist, but not how it has to be implemented. Same goes for describing IoT-A communication. Our system is also much simpler (even though ontologically are similar). There are still things, services, and users, here, but the vast amount of requirements listed as functionalities is not considered appropriate for a nascent system. We have only considered basic things needed for a simple IoT.

VI. CONCLUSION

After screening the literature, we didn’t found architecture that accomplishes ours requirements of simplicity and generality. We therefore conceptually designed simple, secure and general architecture.

On the lowest level (level 1) contemporary link layer protocols should be used with broadcasting mode for transmission of an IoT packet between things and gateway, which has a communication role in our model (level 2). The IoT packet consists of IoT header and IoT payload which should be encrypted with a contemporary symmetric algorithm using thing’s password. Gateways should be used for forwarding messages to and from databases and services located in the cloud (layer 3). Users should observe and control their things with their applications (agents) directly or via the services in the cloud (level 4).

We think that proposed architecture looks promising as a foundation for implementation of tomorrow’s IoT. There is a lot of work ahead to prove its feasibility and usefulness with an example implementation.

REFERENCES

[1] EPCglobal project, http://www.gs1.org/epcglobal, (last accessed 1.4.2015.).

[2] Machine-Type Communications, Third Generation Partnership Project, http://www.3gpp.org/, (last accessed 1.4.2015.).

[3] S. Yang, X. Wen, W. Zheng, and Z. Lu, “Convergence architecture of Internet of Things and 3GPP LTE-A network based on IMS”, Mobile Congress (GMC), Shanghai, pp. 1-7, 2011.

[4] Machine to Machine Communications, European Telecommunication Standards, http://www.etsi.org/technologies-clusters/technologies/m2m, (last accessed 1.4.2015.).

[5] Near Field Communication, http://nfc-forum.org, (last accessed 1.4.2015.).

[6] S. Bendel et al., “A service infrastructure for the Internet of Things based on XMPP”, Pervasive Computing and Communications Workshops (PERCOM Workshops), San Diego, pp. 385-388, 2013.

[7] J. Li, Y. Shvartzshnaider, J. Francisco, R. P. Martin, and D. Raychaudhuri, “Enabling Internet-of-Things services in the MobilityFirst Future Internet Architecture”, IEEE Int. symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM), San Francisco, pp. 1-6, 2012.

[8] K. Toumi, M. Ayari, L. A. Saidane, M. Bouet, and G. Pujolle, “HAT: HIP Address Translation protocol for Hybrid RFID/IP Internet of Things communication”, Int. conf. on Communication in Wireless Environments and Ubiquitous Systems: New Challenges (ICWUS), Sousse, pp. 1-7, 2010.

[9] A. Zakriti and Z. Guennoun, “Service entities model for the internet of things: A bio-inspired collaborative approach”, Int. conf. on Multimedia Computing and Systems (ICMCS), Ouarzazate, pp. 1-5, 2011.

[10] M. Kovatsch, M. Lanter, and S. Duquennoy, “Actinium: A RESTful runtime container for scriptable Internet of Things applications”, Int. conf. on Internet of Things (IOT), Wuxi, pp. 135-142., 2012.

[11] G. Dai, “Design and implementation on SOAP-based things management protocol for internet of things”, Intelligent Control and Automation (WCICA), Beijing, pp. 4305-4308, 2012.

[12] C. Fan et al., “A scalable Internet of Things Lean Data provision architecture based on ontology”, GCC Conference and Exhibition (GCC), Dubai, pp. 553-556, 2011.

[13] H. Hada and J. Mitsugi, “EPC based internet of things architecture”, RFID-Technologies and Applications (RFID-TA), Sitges, pp. 527-532, 2011.

[14] D. Tracey and C. Sreenan, “A Holistic Architecture for the Internet of Things, Sensing Services and Big Data”, Cluster, Cloud and Grid Computing (CCGrid), Delft, pp. 546-553, 2013.

[15] Z. Peng, Z. Jingling, and L. Qing, “Message oriented middleware data processing model in Internet of things”, Computer Science and Network Technology, Changchun, pp. 94-97, 2012.

[16] F. Andreini, F. Crisciani, C. Cicconetti, and R. Mambrini, “Context-aware location in the Internet of Things”, GLOBECOM Workshops (GC Wkshps), pp. 300-304, 2010.

[17] G. Ying, J. Huang, and L. GuanYao, “A tag indexing method in the Internet of Things”, Electronics, Communications and Control (ICECC), Ningbo, pp. 681-685, 2011.

[18] Y. Ding, Y. Jin, L. Ren, and K. Hao, “An Intelligent Self-Organization Scheme for the Internet of Things”, Computational Intelligence Magazine, volume 8, issue 3, pp. 41-53, 2013.

[19] P. Malo, B. Almeida, R. Melo, K. Kalaboukas, and P. Cousin, “Self-Organised Middleware Architecture for the Internet-of-Things”, Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), Beijing, pp. 445-451, 2013.

[20] G. Pujolle, “An Autonomic-oriented Architecture for the Internet of Things”, Modern Computing, Sofia, pp. 163-168, 2006.

[21] J. Zhang et al., “Supporting Personizable Virtual Internet of Things”, Ubiquitous Intelligence and Computing, Vietri sul Mere, pp. 329-336, 2013.

[22] R. Hummen, H. Wirtz, J. H. Ziegeldorf, J. Hiller, and K. Wehrle, “Tailoring end-to-end IP security protocols to the Internet of Things”, Network Protocols (ICNP), Goettingen, pp. 1-10, 2013.

[23] S. Bae, D. Kim, M. Ha, and S. H. Kim, “Browsing Architecture with Presentation Metadata for the Internet of Things”, Parallel and Distributed Systems (ICPADS), Tainan, pp. 721-728, 2011.

[24] N. W. Bergmann and P. J. Robinson, “Server-based Internet of Things Architecture”, Consumer Communications and Networking Conference (CCNC), Las Vegas, pp. 360-361, 2012.

[25] V. Parviainen, A. Yin, A. Romu, and R. Virkkala, “EfiIoT: An efficient software architecture for internet of things”, Industrial Electronics and Applications, Singapore, pp. 709-712, 2012.

[26] S. R. Bhandari and N. W. Bergmann, “An internet-of-things system architecture based on services and events”, Intelligent Sensors, Sensor Networks and Information Processing, Melbourne, pp. 339-344, 2013.

[27] S. K. Datta, C. Bonnet, and N. Nikaein, “An IoT gateway centric architecture to provide novel M2M services”, Internet of Things (WF-IoT), Seoul, pp. 514-519, 2014.

[28] OpenIoT – FP7 project, http://openiot.eu (last accessed 1.4.2015.).

[29] IoT-A – FP7 project, http://www.iot-a.eu (last accessed 1.4.2015.).