Australian Journal of Basic and Applied Sciences, 9(33) October 2015, Pages: 271-277

ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

The Internet of Things Software Architectural Solutions

[1]Mustafa S.Khalefa, [2]Marzanah A.Jabar, [3]Waleed Noori Hussein, [4]Hamid Ali Abed Alasad, [5]Hamzah.F Zmezm

[1]University Putra Malaysia (UPM), Department of Software engineering & Information System ,Faculty of Computer Science &Information technology, 43400 Selangor, Serdang, Malaysia, Computer Science, Education College for Pure Science, Basra University, Iraq.
[2]University Putra Malaysia (UPM), Department of Software engineering & Information System ,Faculty of Computer Science &Information technology, 43400 Selangor, Serdang, Malaysia.
[3]Iraq University College, Basrah, Iraq, departement of computer science and communication.
[4]Computers Sciences Department, Education for Pure Science College, University of Basra, Basra, Iraq.
[5]University Putra Malaysia (UPM), department of computer and communication system engineering, faculty of engineering, 43400 Selangor, Serdang, Malaysia.

ARTICLE INFO

Article history:
Received 3 October 2015
Accepted 31 October 2015

Keywords:
Internet of Things,
IOT, software architectural styles,
quality attributes,

ABSTRACT

The Internet of Things (IoT) is an ambiguous term. There are different definitions for this term, ranging from any system that has sensors and actuators to a solitary interconnected network of physical items. This Paper shows that this term does not give enough information to construct a software architecture on. This is accomplished by taking an inside look at the IoT described in literature as well as the types of applications that exist on the market today and using the concept of software architectural styles to show how different areas in the IoT will need varying styles. This paper continued to classify solutions in the Internet of Things into different classes. The outcomes are that for a subset of the classes there is a reasonable style, however for remaining classes there are still different decisions where more context information is needed

To Cite This Article: Mustafa S.Khalefa, Marzanah A.Jabar, Waleed Noori Hussein, Hamid Ali Abed Alasad, Hamzah.F Zmezm, The Internet Of Things Software Architectural Solutions. ***Aust. J. Basic & Appl. Sci.,*** *9(33): 271-277, 2015*

INTRODUCTION

The Internet of Things (IoT) is an equivocal idea. The term is viewed as equivocal and is for the most part used to describe systems that connect the physical world to the digital world. This is achieved by giving real-world physical objects, which were previously disconnected from any type of network, connectivity to the digital world in various ways (Weimin Wang, 2011). Advancements in the area of small processing units prompting to sensors that can quantify attributes of real world items, actuators that can change the condition of the internet, RFID for distinguishing and finding objects and Cloud technology for computation and data storage all contributed to the current state of the Internet of Things. Applications in the IoT can extend from connecting the human body by use of a wearable sensor to a smart city. solution with multiple sensors distributed throughout the city (Weimin Wang, 2011). The appearance of the Internet of Things carries with it many potential and difficulties. One of the areas of research in the Internet of Things is software architecture. There have been several proposals of reference architectures for the Internet of Things as a single type of system. However, given the ambiguity of the term and the diverse applications regarded as being part of the Internet of Things, it seems unlikely that a one size fits all reference architecture can exist. This paper concentrate on investigating the Internet of Things and deduce a definition in order to bring some clarity to the term. This paper will consider the IoT described in literature as well as the current state of the IoT by looking at different solutions that exist today. The differences and similarities between the vision for and the truth of the Internet of Things will be highlighted. Through discovering the shared factors between the different literature and IoT solutions, it becomes possible to derive a single definition for the term Internet of Things in order to reduce the ambiguity. This paper will contend that there can be no single reference architecture design for the Internet of Things, because the term does not give the context information that is expected to make many outline design. This is an essential explanation to make as it demonstrates the incompatibility of the term Internet of Things and software architecture, i.e.

Corresponding Author: Mustafa S.Khalefa, Marzanah A.Jabar University Putra Malaysia (UPM), Department of Software engineering & Information System ,Faculty of Computer Science & Information technology, 43400 Selangor, Serdang, Malaysia.
Phone: 0060176484073; E-mail: mustafakhalefa@gmail.com, marzanah@upm.edu.my

the term must not be utilized to describe a type of system when discussing about software architecture. Future papers should be more particular about which sort of solutions their reference architectures are proposed for in order for them to be useful. Therefore, a set of IoT solutions will be assembled into classes in order to illustrate how various the IoT solution space can be. This is done by investigating a few IoT solutions and illustrating the differences and similarities between them. This paper won't give a complete set of classes as this is out of scope and cannot be proven to be complete. This paper will give prove that this objective of achieving such a one size fits all" architecture is not possible for the Internet of Things, thanks to the ambiguity of the term, diversity in applications and continuous evolution of the concept. This will be done by utilizing software architectural styles and evaluating the quality effects on a system using them.

IOT:

The IoT has turned into a popular expression much like Big Data" in the sense that it is used frequently but seems to be ambiguous to the users of the term (Geng Yang, 2010). For many of the papers written about the Internet of Things, two qualities have been mentioned the most. The first one explained that the IoT is a network. It is always referred to in the singular form, which may recommend that there is one IoT only. The second most mentioned characteristic in the definitions refer to the things" in the IoT. These are ordinary physical items. These physical objects are to be installed with innovation so that they can be associated with the IoT, have their state measured and changed, are uniquely identifiable and can communicate with each other to achieve greater value and service (DiRomualdo, A. and V. Gurbaxani, 2002). The goal of accomplishing greater value and service is extremely broad. This is due to the fact that the variety in applications for the IoT is enormous. This greater value and service provided by the IoT is either the formation of new functionality previously not possible without the network of physical objects or the increase of the quality of existing processes with the help of the IoT (Yong, A., 2014). The oddity of the IoT is available in the name. The term Things" refers to the everyday physical objects that will now become connected (Yong, A., 2014). This will make new kind of data accessible that was previously not possible. The term Internet is utilized to represent the interconnection between these networks of heterogeneous objects (Zhang, Baoquan, 2011). The IoT also has to be dynamic, since nodes will be added and removed constantly from this network. New sorts of devices will rise and the IoT has to be able to handle this change. The IoT should be self-conjuring, always adapting to this change. The terms global, world-wide, ubiquitous are mentioned for the IoT. The IoT has to be accessible everywhere. With the Internet being as pervasive and ubiquitous as it is today, this is not an unreasonable requirement. Table 1 shows the characteristics pervasive system (DiRomualdo, A. and V. Gurbaxani, 2002).

Table 1: Characteristic of pervasive system

characteristics	Pervasive system
Invisible by design	Pervasive systems are not explicitly there. They are often integrated common objects
Networked	Devices are interconnected by a seamless communication infrastructure
Many-to-many	As opposed to one-to-one or one-to-many relationships.
Always on	Devices do not need to be actively switched before interaction can be had.
Distributed	The computing intelligence is a combined computing effort of multiple devices
Context-aware	Can measure their environment and is aware of other pervasive devices in their vicinity.
Adaptive	The actions of the system are triggered by implicit actions rather than explicit user interaction.
Natural human interface	People should not need to think about how to inter- act with the system, this should be natural through speech, touch or movement.

There is much more written about the Internet of Things, however this paper will concentrate on the software architecture perspective. Hence, it is sufficient to realize what is generally meant by the Internet of Things in literature. In any case, we will also require information on what the quality non-functional requirements can be for the IoT.

IoT Reality:

This section describes the present state of the IoT by looking at the solutions that are available because there is a major variety in IoT solutions, the data set must be illustrative of all types" of solutions. Postscapes categorizes the IoT into six domains (Yong, A., 2014). These categories are the connected home, connected body, associated retail, associated transportation, smart city and industrial application (Zhang, Baoquan, 2011). These categories almost correspond completely with the domains mentioned in the IoT research, aside from the IoT also includes E-Health and Smart Energy as an IoT domain. The E-Health domain covers with some of the other domains, so this will also be overlooked. From each domain, three solutions will be taken and dissected in depth. The detail in which the investigation is done can contrast between solutions, as some are less likely to share information than others. An overview of the solutions analysed can be seen in Table 2 (Yong, A., 2014).

Table 2: Overview of domains and IoT solutions

Domain	Solutions
Connected home	SmartThings
Connected body	Zebra Motion works
Connected retail	Scanalytics oor sensors, S5 Electronic Shelf la-bels, Nomi Brickstream live
Connected transportation	Weather Cloud, Truvolo Car Solution, Veniam Vehicular Networking
Smart city	Bitlock bycicle lock, Array of Things, Enevo waste collection
Industrial application	Farmobile Fleet Management, Condeco Workspace Occupancy Sensor, DAQRI Smart Helmet

The methodology for analysing the solutions was done systematically by looking at the following variables:

1. Identify the Physical Entity being measured. This is done to confirm that the system can be classified as an IoT solution.
2. Identify the attribute(s) of the physical element that is being measured. This is also done to check that the system can be classified as an IoT solution.
3. Identify if the sort of IoT Connector(s) present in the solution.
4. Identify the components and the topology of the network.
5. Illustrate the topology in a diagram showing the relationship between parts (one-to-one, one-to-many, many-to-many).
6. Illustrate the direction of messages passed between the components in a diagram
7. Identify the area of application logic and data storage for the solution. The application logic and data storage locations do not refer to logic and data needed to network between nodes, rather it refers to the logic and data that are specific to the solution.
8. Identify the client interaction possibilities of the solution.
9. Make an estimation of the scalability requirement of each component. The scale can be fixed or potentially increasing.
10. Identify the Internet-Dependency of the solution.

The objective of this investigation is to demonstrate that the physical entity and measured and changed attributes can be distinguished gather important characteristics of IoT solutions to be used for recommending software architectural styles.

A research done to outline the imperative requirements for the IoT based on the opinions of stakeholders for the IoT was conducted by the Internet of Things Architecture (IoT) institute (Wegmann, A., *et al*., 2001). The most important requirements found were Interoperability, evaluability, performance, scalability, availability, resiliency, security and privacy. Different studies support the notion that these are important requirements for the IoT. In this paper, scalability will be considered as major aspect of performance while resiliency will be seen as part of availability.

The purpose behind this decision is because in software evaluation methods, these properties are also merged in this way. Interoperability is mentioned as a sub-characteristic of compatibility in the ISO standard. Compatibility is the ability of a system to exchange information with other systems while sharing the same hardware or software environment. Interoperability is defined as the ability of two or more systems to exchange information and also use the information that has been changed. Evaluability is not mentioned in the ISO-standard, but rather can be portrayed as a combination of modularity and mod ability .Performance is characterized by three sub-characteristics. These are time-behaviour, resource utilization and capacity. The architecture of a system can be have an effect on all three of these sub categories. Scalability can be mapped to capacity, which is described as the degree to which the maximum limits of a system meet requirements. For a system to meet the required work load, it might have to be scalable depending on the context. Availability is mentioned in the ISO standard as a sub characteristic of Reliability. Resiliency can be seen as a combination of fault tolerance and recoverability. Security and all of its sub-characteristics is mentioned in the ISO standard. These are Confidentiality, integrity, non-repudiation, authenticity and accountability. Privacy is not a part of the standard. While confidentiality covers a part of privacy, there are also other aspects of privacy that need attention in the IoT.

Solution Classes for IOT:

The initial step into demonstrating the impact of software architectural styles in the IoT while simultaneously representing that the IoT should not be seen as one system is to separation it into classes. In order to perform this, a number of classifying attributes must be identified. The classification was made with the following constraints in mind:

1. The classification is made essentially using the dataset of 15 solutions analysed.
2. The architecture of these solutions might not be used as classifying attributes. This incorporates published architectures as well as the component topology that can be gotten from the description of the solutions. If this were to be included, then the choices for possible styles would be restricted.
3. No quality attribute requirements will explicitly be used as classifying attributes (http://iso25000.com/index.php/en/iso-25000standards/iso-25010).

Classification is done by comparing the solution for each other. There are of course many ways in which each solution differs from the next, however the subset are referred as shown in Table 3:

Table 3: The Classification

User interface on devices.	The devices at the edge of the network can have a client interface or be controlled by an application on another device.
Sensors and Actuators.	The solutions can either have only sensors or additionally have actuators.
Devices connected directly to a network.	The devices can be specifically associated to a network or might make use of different approaches to interact with the system.
Device is stationary or mobile.	This can have an effect on accessibility. If a device is stationary, it will probably have a stable connection if this is required.
User interface application.	Some solutions do not come with a user interface application.
Data per user or collective.	Some solutions provide data gathered for a specific user while other systems provide data collected by all sensors for all users.
Number of devices per user.	The values can be one, one to few (constrained), numerous.
Devices battery or plugged into power outlet.	If the devices use batteries it might be in the best interest to limit computation and communication to the devices as much as possible.
Devices constrained to a location	If all of the devices are constrained to a location for example, a home or a workspace, then certain design choices can be made in order to improve scalability.
Autonomous behaviour	Solutions can show independent conduct, by this we are particularly inspired in actuators being controlled without the use of human Interaction.

Not all of these characteristics are used in the classification. It is also the case that some of these attributes are dependent on each other. For instance, the constrained to a location attribute is false if the devices are not stationary. This will be considered in the classification. Figure 1 shows the resulting decision tree after several revisions.

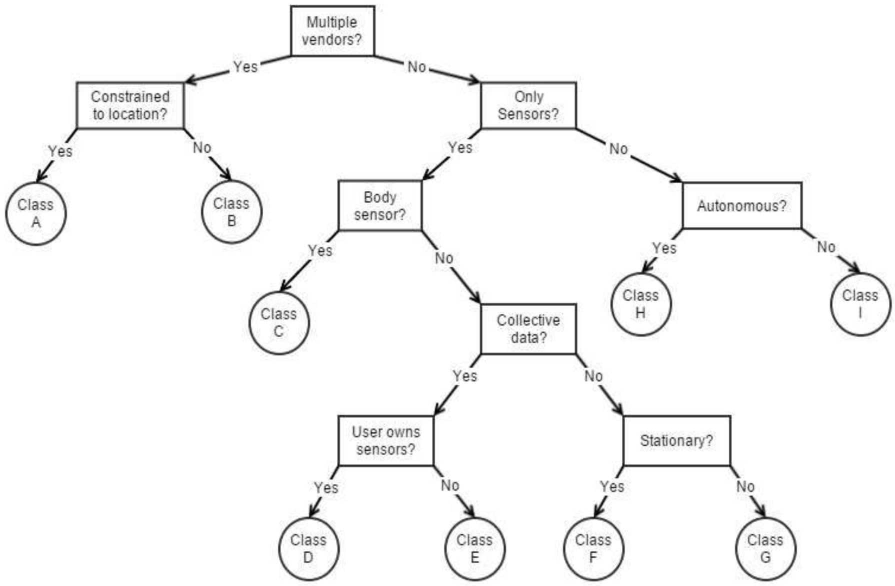

Fig. 1: IoT Classification Decision Tree

Note that a decision tree won't be viewed an ideal classification technique for this reason, as a few of classifying attributes can be put at different areas of the tree since they are not dependent on each other. The stationary attribute could also be placed to split class G or H for example. Furthermore, it is not possible to figure the accuracy of this classifier for the whole solution space as this would require a dataset containing all, or at least a significant amount, of solutions. However, this decision tree ought to be sufficient to give a set of classes that can be utilized to show the variety of solutions and the need for different styles. The goal is not to provide a complete set of classes for the IoT. This particular structure gives us a view into two dimensions and four major categories of IoT solutions. The dimensions correspond to the two nouns present in the term IoT, namely Internet and Things. As illustrated in figure 2, solutions located on the left side of the tree are responsible for creating the interconnected network of things by providing interoperability between multiple solutions when this is necessary. The right side of the tree contains the independent solutions that have sensors and/or actuators that connect the physical world to the digital world. The solutions on the right side of the tree can be viewed as the building blocks of the IoT, however the solutions at the left side can be viewed as the glue that will paste them together.

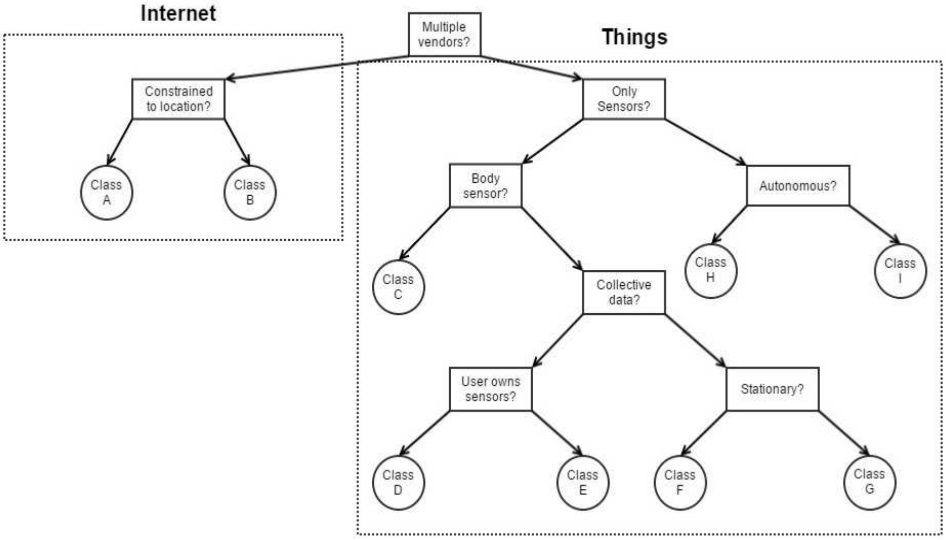

Fig. 2: IoT solution Dimensions

Classes:

In this section classes will be given names Table 4 shows a labelled set of IoT classes that will be used

Table 4: labeled set of IoT classes that will be used

Classes	Characteristics
Location Constrained Heterogeneous Devices.	This class is ordered by solutions that give interoperability between numerous solutions that contain device in the same location, for example, a smart home or work office.
Location Free Heterogeneous Devices.	This class also gives interoperability between solutions, aside from the devices can be found anywhere. This means that a central point have to be located on a server possibly in the Cloud to give a single point of interaction.
Body Sensors.	Characterized by the one-to-one relationship between users and devices, this class contains solutions that monitor and observes measurements of the user.
Active User Collective Data Solutions.	This solution intends to gather data from multiple sensors and provide an examination on the whole set of data as a whole. Users are considered active as they contribute to the data set actively through sensors that they own
Passive User Collective Data Solutions.	The difference between this class and the previous is that the users of the data are not owners of the sensors. This means that the number of sensors is entirely in control of the party that owns the solution, meaning scalability only has to be handled based on the number of users.
Stationary Homogeneous Sensors.	This class contains solutions that have sensors that are stationary and are of the same type.
Mobile Homogeneous Sensors.	These solutions have the same objective as the previous class with exception that the sensors can be in movement, meaning that giving availability turns into a more essential requirement. Solutions that fall into this class are the Far- mobile, DAQRI smart helmet, Truvolo and Veniam solutions.
Smart Systems.	The term smart is used a lot these days to describe any alternative version of a device or system that gives some automation. In this classification we utilize it to describe independent solutions that are able to use data and logic and convert it into decisions that can prompt to actuator commands without human intervention.

Software Architectural Styles in the Internet of Things:

A software architectural style is a marked set of components and connectors, and a set of constraints on how they can communicate (Bauer, M., *et al*., 2013; Garlan, D. and M. Shaw, 1994). These limitations can be topological, for instance not allowing cycles, or it can regard execution semantics. The latter refers to the meaning of such an interaction between two components, which could be a procedure call or a notification for instance. All styles accompany with trade-offs, explicitly mentioning which quality attributes are gained and which are given away, however this also depends on the context of the system to be built. The software architectural styles that will be considered in this paper are Client-Server, Peer-to-Peer, Pipes-and-Filters, Event-Based, Publish-Subscribe, Service-Oriented, REST, Layered and Microkernel. There different styles that exist, however these probably the most common and well documented ones. For the mapping we will distinguish what the quality attribute requirements are for each class. The architectural styles provide variations in how these requirements are fulfilled by the architecture, which will allow us to compare them with each other as shown in Table 5.

Table 5: The architectural styles

Interoperability.	For interoperability the necessities could either be primary or secondary. We have seen enough cases of solutions where interoperability is not mentioned at all, however for this investigation we will categorize these solutions as having interoperability as a secondary requirements.
Evolvability.	Is about reducing the cost of change to the system. For every class of solution we will show some of the likely changes to happen. The choice in style will dictate how and where these changes will occur and thus how evolvable the architecture is.
Performance.	We will consider latency, through put, power consumption/energy efficiency, bandwidth efficiency and scalability as characteristics that define performance in the IoT. These will all be affected by the choice of architectural style. Latency can be measured by the number of hops needed to reach the destination.
Availability.	We can make an estimation of how much effect a single device being inaccessible could be. We can also identify single-points of failure inherent in the classes and their goals.
Security.	Security is always a priority need. For this purpose we will not make an estimate on the requirement for this attribute, however we will indicate to it later to see if the select of architectural style has an impact.
Privacy.	Some solutions, such as the ones that have collective open data, have less of a privacy requirements than other systems.

Mapping:

This section presents a format with which the mapping will be conducted and the actual mapping itself. This will make sure that the investigation can be performed in a systematic way, as well as making sure that all possibilities have been considered. The following Table 6 shows the format that was used.

Table 6: The Format

For each category	Verdict	For each class:	For each style
Description		Description	Description
		Functional Requirement(s)	Quality Attribute Effects
		Quality Attribute Specifications	

This indicates that for every one of the four types it will look at the classes and what the effects of software architectural styles are on them. However, this paper focuses on quality attributes, a list a few functional requirements will be mentioned as this will help get a better view of what functionality the system should provide which can help to eliminate styles that are not suitable because their constraints are not compatible. The quality attribute specification shows what the specific quality attributes mean in the context of that class and how much of a priority they are. It must be noted that all styles can be done to create any type of system, however this has an effect on the quality attributes that the system will exhibit. An example is using a Peer-to-Peer style for data aggregation. This is conceivable, however it will not execute as well as using a more centralized style where the data comes together at one point. At the end of each class description there will be a verdict indicating which classes are the best fit.

Conclusion:

The concept of the Internet of Things is covered in ambiguity. There is contrast between the definitions provided in literature and the solutions that are available right now. The aftereffect of this is that the IoT in literature has numerous quality attribute requirements that are not important to all solutions that are marked as IoT. To further expand the confusion, the investigation done on a set of Internet of Things solutions that exist at the moment reveals a great variety of goals, requirements and implementations.

REFERENCES

Weimin Wang, 2011. The research and development of the Internet of Things technology. Information network security, 03: 53-56.

Geng Yang, 2010. The characteristic and key technology of the Internet of Things. The academic journal of Nanjing University., 30(4): 20-29.

Zhang, Baoquan, 2011. Evaluation on Security System of Internet of Things Based on Fuzzy-AHP Method E Business and E -Government (ICEE), International Conference.

DiRomualdo, A. and V. Gurbaxani, 2002. “Strategic Intent for IT Outsourcing.” Sloan Management Review, 39(4): 67 80.

Yong, A., 2014. “Outsourcing Market View: What th Future Holds.” Presentation at Gartner IT Services and Sourcing Summit, Las Vegas.

Hoepman, J.H., 2012. In things we trust? towards trustability in the internet ofthings. Constructing Ambient Intelligence, pp: 287-295.

Ahamed, S.I. Yau S.S. Buduru A. Addo, 2014. I.D. Reference architectures for privacy preservation in cloud-based iot applications. International Journal of Services Computing, 2: 65-78.

Wegmann, A., O. Preiss and J. Wong, 2001. On quality attribute based software engineering. Euromicro Conference, 114{120, 2001}.

http://iso25000.com/index.php/en/iso-25000standards/iso-25010.

Bauer, M., M. Fiedler, T. Kramp, R. Van Kranenburg, S. Lange, A. Bassi and S. Messner, 2013. Enabling things to talk. Designing IoT solutions with the IoT Architectural Reference Mode.

Garlan, D. and M. Shaw, 1994. An introduction to software architecture.

Chui, M., P. Bisson, J. Woetzel, R. Dobbs, J. Bughin, D. Aharon, J. Manyika, 2015. The Internet of Things: Mapping the value beyond the hype.