University of Koblenz-Landau

Master Thesis

Internet of Things (IoT) Development Platforms – A Case of IBM Bluemix

Pawan Kumar
Master Web Science
Matriculation Number: 214202799

Supervised by
Professor Dr Susan P. Williams
Ms. Clara Greeven

September 25, 2016

Universitätsstraße 1
56070 Koblenz-Metternich
Germany
pkumar@uni-koblenz.de

Erklärung

Hiermit bestätige ich, dass die vorliegende Arbeit von mir selbständig verfasst wurde und ich keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet–Quellen – benutzt habe und die Arbeit von mir vorher nicht in einem anderen Prüfungsverfahren eingereicht wurde. Die eingereichte schriftliche Fassung entspricht der auf dem elektronischen Speichermedium (CD-Rom).

	Ja	Nein
Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.	☐	☐
Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.	☐	☐

....

(Ort, Datum) (Unterschrift)

Table of Contents

Contents

List of Figures

List of Tables

Abbreviations

AMQP	Advanced Message Queuing Protocol
ARM	Architectural Reference Model
AWS	Amazon Web Services
CLI	Command Line Interface
CoAP	Constrained Application Protocol
CPPS	Cyber-Physical Production Systems
DSR	Design Science Research
DTLS	Datagram Transport Layer Security
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
IaaS	Infrastructure as a Service
ICT	Information and Communication Technology
IETF	Internet Engineering Task Force
IoT	Internet of Things
IoT-A	Internet of Things - Architecture
ISO	International Organization of Standardization
M2M	Machine to Machine
MQTT	Message Queue Telemetry Transport
NIST	National Institute of Standards and Technology
OASIS	Organization for the Advancement of Structured Information Standards
PaaS	Platform as a Service
REST	Representational State Transport
RFID	Radio Frequency Identification
SaaS	Software as a Service
SASL	Simple Authentication and Security Layer
SDK	Software Development Kit
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
UML	Unified Modelling Language

Zusammenfassung

Das Ziel dieser wissenschaftlichen Arbeit ist es, verschiedene vorhandene cloud-basierte Internet of Things (IoT) Entwicklungsplattformen zu untersuchen und hier im Speziellen eine Plattform (IBM Watson IoT) anhand eines Anwendungsfallszenarios detailliert zu untersuchen. Bei IoT handelt es sich um eine aufkommende Technologie mit der Vision, die virtuelle Welt (z.B. Clouds, soziale Netzwerke) und die physikalische Welt (z.B. Geräte, Autos, Kühlschrank, Menschen, Tiere) durch die Internettechnologie miteinander zu verknüpfen. Beispielsweise kann das IoT-Konzept von "smart cities", welche das Ziel verfolgen, die Produktivität und die Geschäftsentwicklung sowie die sozialen und kulturellen Angebote in der Stadt zu verbessern, durch die Nutzung von Sensoren, Aktuatoren, Clouds und mobilen Geräten erreicht werden (IEEE, 2015). Ein Sensor (z.B. ein Temperatursensor) in einem Gebäude (globale Welt) kann Echtzeitdaten an die IoT Cloud-Plattform (virtuelle Welt) senden, wo sie überwacht, gespeichert und analysiert werden oder eine Aktion auslösen können (z.B. das Kühlsystem in einem Gebäude anschalten, wenn die Temperatur eine bestimmte Grenze überschreitet). Obwohl IoT viele Möglichkeiten in verschiedenen Bereichen schafft (z.B. Transportwesen, Gesundheitsversorgung, verarbeitende Industrie), bringt es auch Herausforderungen mit sich, wie z.B. die Standardisierung, Interoperabilität, Skalierbarkeit, Sicherheit und Privatsphäre. In diesem Bericht werden IoT Konzepte und verwandte Schlüsselprobleme behandelt.

Der Fokus dieser wissenschaftlichen Untersuchung liegt in dem Vergleich verschiedener cloud-basierter IoT Plattformen, um die geschäftlichen Aspekte und die technischen Funktionen zu verstehen, die diese bieten. Die cloud-basierten IoT Plattformen von IBM, Google, Microsoft, PTC und Amazon wurden dabei untersucht.

Für die Ausgestaltung dieser Arbeit wurde die Design Science Research (DSR) Methode verwendet; für die Nachbildung des Echtzeit-IoT Systems wurde die IOT-A modellig Methode verwendet.

Der Vergleich verschiedener cloud-basierter IoT Entwicklungsplattformen zeigt, dass alle untersuchten Plattformen einfache IoT Funktionen bereitstellen, wie z.B. die Verbindung von IoT Geräten und der cloud-basierten IoT Plattform, das Sammeln von Daten von IoT Geräten, die Datenspeicherung und die Datenanalyse. Jedoch ist die IBM IoT Plattform den anderen Plattformen gegenüber aufgrund des integrierten Laufzeitsystems im Vorteil; dies macht sie zudem entwicklerfreundlich. Aus diesem Grund wurde die IBM Watson IoT für Bluemix für die weitere Untersuchung ihrer Einsatzmöglichkeiten ausgewählt. Das Angebot von IBM Watson IoT für Bluemix beinhalten Analytik, Risikomanagement, Verbindungs- und Informationsmanagement. Es wurde ein Anwendungsfall implementiert, um die Einsatzmöglichkeiten der IBM Watson IoT Plattform einzuschätzen. Die digitalen Artefakte (d.h. Anwendungen) wurden entwickelt, um die IoT Lösung von IBM zu bewerten. Die Ergebnisse zeigen, dass IBM eine skalierbare und entwickler- und einsetzungsfreundliche IoT Plattform bietet. Die enthaltene kognitive, kontextuelle und vorhersehbare Analytik erlaubt eine

vielversprechende Funktionsweise, die Einblicke auf Basis der IoT Daten gewährt, die durch Sensoren oder andere IoT Geräte übertragen werden.

Abstract

The purpose of this research is to examine various existing cloud-based Internet of Things (IoT) development platforms and evaluate one platform (IBM Watson IoT) in detail using a use case scenario. Internet of Things (IoT) is an emerging technology that has a vision of interconnecting the virtual world (e.g. clouds, social networks) and the physical world (e.g. device, cars, fridge, people, animals) through the Internet technology. For example, the IoT concept of smart cities which has the objectives to improve the efficiency and development of business, social and cultural services in the city, can be achieved by using sensors, actuators, clouds and mobile devices (IEEE, 2015). A sensor (e.g. temperature sensor) in the building (global world) can send the real-time data to the IoT cloud platform (virtual world), where it can be monitored, stored, analysed, or used to trigger some action (e.g. turn on the cooling system in the building if temperature exceeds a threshold limit). Although, the IoT creates vast opportunities in different areas (e.g. transportation, healthcare, manufacturing industry), it also brings challenges such as standardisation, interoperability, scalability, security and privacy. In this research report, IoT concepts and related key issues are discussed.

The focus of this research is to compare various cloud-based IoT platforms in order to understand the business and technical features they offer. The cloud-based IoT platforms from IBM, Google, Microsoft, PTC and Amazon have been studied.

To design the research, the Design Science Research (DSR) methodology has been followed, and to model the real-time IoT system the IOT-A modelling approach has been used.

The comparison of different cloud based IoT development platforms shows that all of the studied platforms provide basic IoT functionalities such as connecting the IoT devices to the cloud based IoT platform, collecting data from the IoT devices, data storage and data analytics. However, the IBM's IoT platform appears to have an edge over the other platforms studied in this research because of the integrated run-time environment which also makes it more developer friendly. Therefore, IBM Watson IoT for Bluemix is selected for further examination of its capabilities. The IBM Watson IoT for Bluemix offerings include analytics, risk management, connect and information management. A use case was implemented to assess the capabilities that IBM Watson IoT platform offers. The digital artifacts (i.e. applications) are produced to evaluate the IBM's IoT solution. The results show that IBM offers a very scalable, developer and deployment friendly IoT platform. Its cognitive, contextual and predictive analytics provide a promising functionality that can be used to gain insights from the IoT data transmitted by the sensors and other IoT devices.

Chapter 1. Introduction

This chapter provides an introduction to the thesis topic. The structure of document, research background, motivations, research objectives and related research questions are discussed in this chapter.

1.1 Background

In the manufacturing industry, the next development stage in the organisation and management of the entire value chain process is referred to as Industry 4.0 or 'fourth industrial revolution' (Deloitte, 2014). This concept is used across Europe, especially in Germany's manufacturing industry. A similar concept is called the 'Internet of Things', 'Internet of Everything' or the 'Industrial Internet' in the English-speaking world. Figure 1 shows the evolution of Industry over time. As Industry evolves, the degree of complexity also increases (Deloitte, 2014).

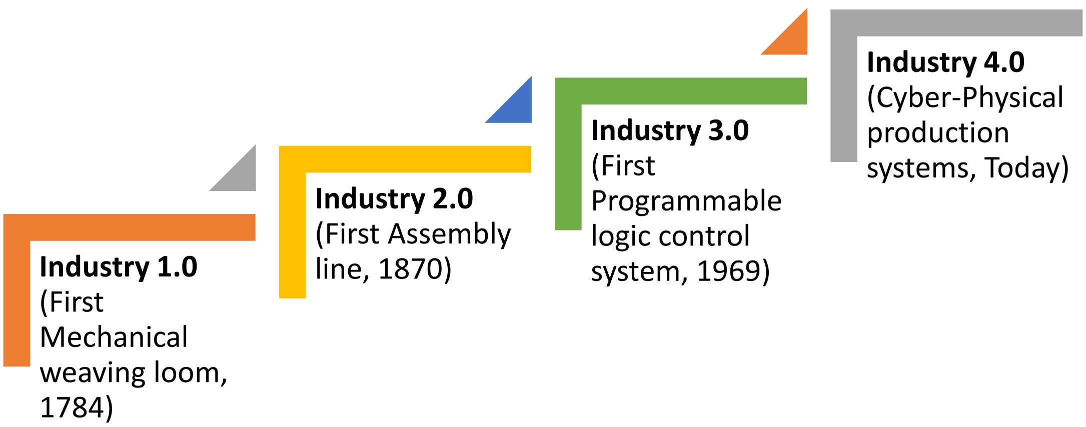


Figure 1: Evolution of industry, adapted from (Deloitte, 2014, p. 3)

Industry 1.0: First mechanical weaving loom (1784). (Mechanical production using water and steam power).
Industry 2.0: First assembly line (1870). (Mass production using electrical energy).
Industry 3.0: First programmable logic control system (1969). (Electronics and IT applications).
Industry 4.0: Cyber-Physical production systems (CPPS) (Today), (Merging real and virtual world).

CPPS are online networks of social machines, which interact with each other through a network. CPPS provide the basis for creating the Internet of Things (Deloitte, 2014). There is still no official definition of the Internet of Things (IoT) and organisations working in the field of IoT are creating their own definition (IEEE, 2015).

According to IEEE (2015), The Internet of Things (IoT), in general, is a system that consists of networks of sensors, actuators, and smart objects. We are in an era of change, which is being entered by new innovative services and new business actors (Andersson & Mattson, 2015; Orlikowski & Scott, 2008, p. 447). The IoT field stretches across different domains including smart healthcare, smart cars, smart enterprise, smart energy, smart transport, smart city (smart street lighting, waste management, smart parking, smart pest control, driver-less vehicles, water monitoring, journey planning, alerts) (postscapes, n.d.).

1.2 Motivation

Various studies (Evans, 2011; Ericsson, 2016; Gartner, 2013) forecast that there will be a very large number of connected devices by 2020 (see Figure 2). These devices will be generating a huge amount of data, and this will result in a growing revenue from IoT products and services. Cisco's prediction of 50 billion connected devices in 2020 might be a hype, but many companies are working in developing IoT solutions (i.e. hardware and software) that could make it a reality.

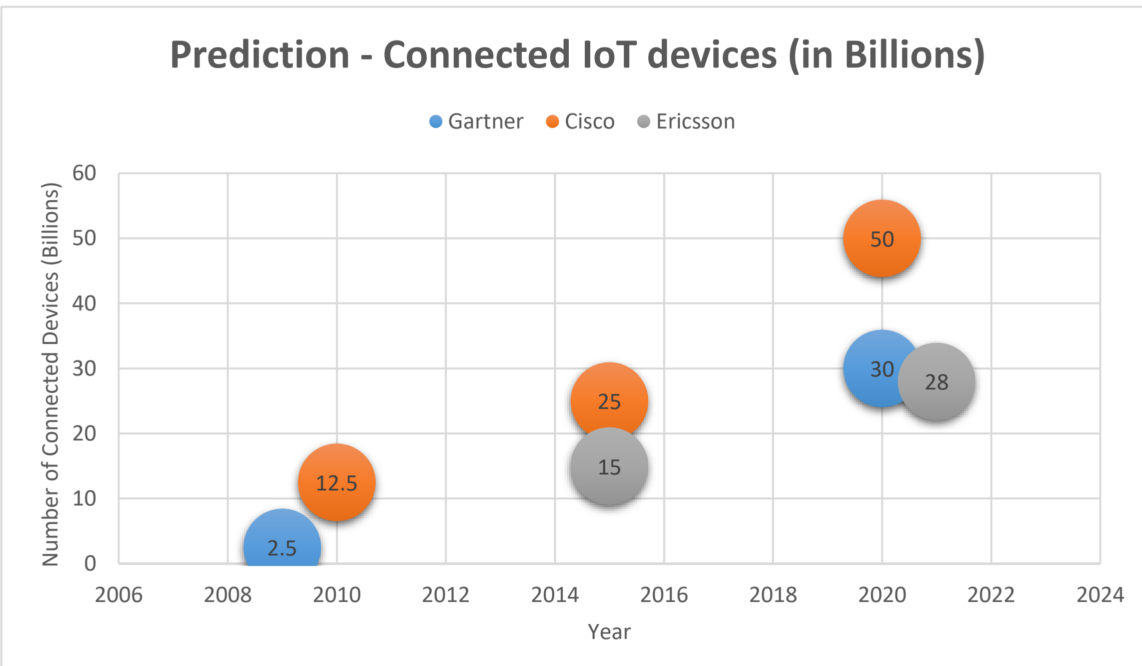


Figure 2: Connected devices prediction, adapted from (Evans, 2011; Gartner, 2013; Ericsson, 2016)

Presently, different standard bodies (see section 3.2), as well as open standards, are working in the IoT field of studies. In 2009-10, the IOT-A (Internet of Things Architecture) project was started to define an Architectural Reference Model (ARM) for the Internet of Things, and in 2013-14, they have delivered the final draft D1.5. A number of other standards, for example, IEEE and ISO/IEC are also working in the same area of defining the architecture for the Internet of Things. Big names in information technology, i.e. Microsoft, Amazon, IBM, and Google etc., have already launched cloud-based platforms for IoT. Along with this, different organisations are also involved in manufacturing smart devices, sensors, actuators, etc. to empower the idea of IoT.

Despite the vast opportunities in the IoT field, there are also many challenges (refer section 3.3). The field of IoT, both in theory and practice, is still in its early days and is still experiencing a number of problems.

There is uncertainty about IoT due to the various standards (i.e. IOT-A, IEEE, ISO/IEC, etc.) and standards bodies that are working independently in the IoT field of study, different IoT protocols (i.e. MQTT, AMQP etc.) that could result in interoperability issues, and many IoT development platforms which deliver the proprietary IoT solutions.

There is a strong need to reduce uncertainty and to capitalize on the many possibilities that IoT offers. This signals that there is a lot of work still to be done. Through this thesis, a relevant contribution is to be made towards a better understanding of IoT development platforms.

1.3 Research Objective and Research Questions

The main aim of this research is to examine existing IoT development platforms and select one platform to study in-depth the feasibility of its IoT solution.

To achieve the research aims, the following research objectives (RO) have been identified through literature study. The research questions (RQ) corresponding to each research objective are also discussed below.

RO-1: Understanding the meaning of IoT and different terminologies used in the field of IoT

- **RQ-1a:** How can IoT be defined?
- **RQ-1b:** What are common terms and terminologies in the field of IoT, and how can they be defined?
- **RQ-1c:** What are the challenges and issues (i.e. technical (standards, protocols), social (business model, ecosystems)) related to IoT?

RO-2: Examine and compare existing IoT Platforms.

RQ-2a: What are existing IoT development platforms, and what are their business model (i.e. cost structure, customer segments, etc.) and technical model (i.e. database, applications, services, etc.)

RQ-2b: What general capabilities do these existing IoT development platforms offer and what constraints do they have?

RO-3: Examine one IoT Platform from the already examined platforms for in-depth evaluation.

RQ-3a: What technical capabilities and requirements does the selected IoT platform offer and how does it enable the effective development of an IoT service?

RO-4: Show a use case of the selected IoT platform in a particular IoT domain.

RQ-4a: How can the capabilities of the selected IoT development platform be effectively represented through one use case in a particular IoT domain?

The Design Science Research (DSR) methodology is used to design this research. The DSR steps are followed to answer the research questions discussed above. The research methodology is discussed in the next chapter.

1.4 Document Structure

Rest of the document is organised as follows.

Chapter 2 explains the research methodology followed to design the current work. The research steps used to achieve the research objectives and to answer the research question is also explained in this chapter.

Chapter 3 examines the literature to understand the concept of IoT. The IoT domain and scope, ecosystem, challenges, and various IoT communication protocols are also explained in this chapter. The various research papers, standards, available smart devices, and related concepts were studied to understand the different terms and concepts used in the field of Internet of Things.

Chapter 4 presents and analyses different existing cloud-based IoT development platforms. The reviewed platforms are compared with each other using general, business and technical criteria. This chapter constitutes the major part of this thesis and explains the technical capabilities of each studied system in detail. IBM Watson IoT for Bluemix platform is selected for in-depth evaluation. This chapter also explains the various components of the Bluemix platform and how IoT devices can be connected to the Bluemix's IoT solution.

The evaluation of the IBM Watson IoT for Bluemix platform, discussed in Chapter 5, is done by implementing a use case scenario. Actual sensors, actuators, mobile devices are used to implement the use case system using IBM's IoT solution.

Chapter 6 concludes the research work and explains the future possibilities in the area of Internet of Things.

Chapter 2. Research Methodology

The design science research approach has been adopted to address the research objectives and related research questions discussed in section 1.3.

2.1 Design Science Research (DSR)

Design means to invent something. Research can be designed by following appropriate steps. According to Hevner and Chatterjee (2010), Design science research is defined as follows:

“Design science research is a research paradigm in which a designer answers questions relevant to human problems via the creation of innovative artifacts, thereby contributing new knowledge to the body of scientific evidence. The designed artifacts are both useful and fundamental in understanding the problem.” (p. 5).

As stated in the definition above, the design science research is applied to create artifacts so as to get the knowledge about the research problems. In 1995, March and Smith (March & Smith, 1995) presented a design science framework. According to the authors, design science consists of two activities: *Build* and *Evaluate*, and it generates four types of products: *Construct*, *Models*, *Methods* and *Implementations*. Hevner et al. (2004) have explained the four products or artifacts as follows: The language to define and communicate the problems and solutions is provided by the constructs, the connection between problem and solution is represented by models, the processes to solve the problems can be defined by methods, and instantiations represent the implemented and prototyped systems. The approach followed by the authors Hevner et al. (2004) is restricted to the constructive research.

In a notable work by Vaishnavi and Kuchler Jr. (2008), the authors have described the steps of design science process and the output expected in each step (Figure 3).

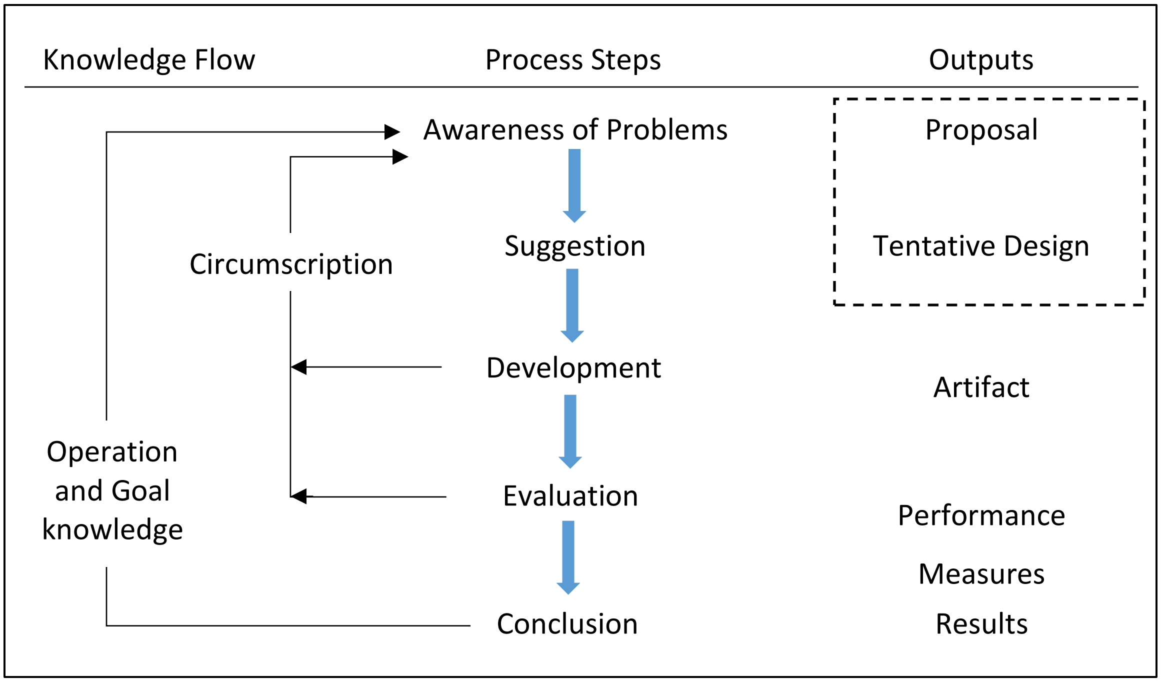


Figure 3: The general methodology of DSR, adapted from (Vaishnavi & Kuechler, 2008, p. 20)

Vaishnavi and Kuchler Jr. (2008) present five steps of design science process: *Awareness of the problem*, *Suggestion*, *Development*, *Evaluation* and *Conclusion*. These steps are explained in detail below (Vaishnavi & Kuechler, 2008, p. 20):

- **Awareness of problem:** Awareness can come from new development in industry or in a reference disciple. The problem can be identified through literature review, experience in practice, conversation with colleagues, etc. The main purpose of this step is to identify and define the problem.
- **Suggestion:** It is a creative step in which new functionalities are visualized based on an existing and new elements. Suggestions are drawn from existing knowledge or theory base for the problem area, appropriate research methodology, etc. To identify potential suggestions, an existing literature study or explanation research study can be very helpful. In this research, cloud-based IoT development platforms have been analysed and compared on the basis of their existing functionalities (see Chapter 4).
- **Development:** This is the major design part in design science research. The artifacts developed can be abstract in nature: construct, models or methods, or it can be tangible in the form of computing software and hardware. In this research, the software artifacts (i.e. barn climate monitor system, barn device control system, cow tracking system) are developed using the IBM Watson and IoT devices (see Chapter 5). The IOT-A modelling approach has been followed to represent the connection between different components used to implement the case scenario.
- **Evaluation:** In this step partial or fully successful implementations are evaluated using action research, controlled experiments, simulations, or scenarios. A use case scenario (cow

farmhouse) is implemented using the software artifacts developed in Development step to evaluate the IBM Watson IoT for Bluemix platform.

- **Conclusion:** This step indicates the termination of the design project.

Hevner (Hevner A. R., 2007) further explained the three cycle view (Figure 4) of design science research based on his previous work. The design science research is initiated by an application domain. The relevance cycle can be iterated as required, and it acts as a bridge between the application domain and the design science activities. The design science and knowledge base are connected by the rigor cycle. The design cycle connects activities to build and evaluate artifacts. According to the authors (Hevner & Chatterjee, 2010) these three cycles must be present in a design science research project.

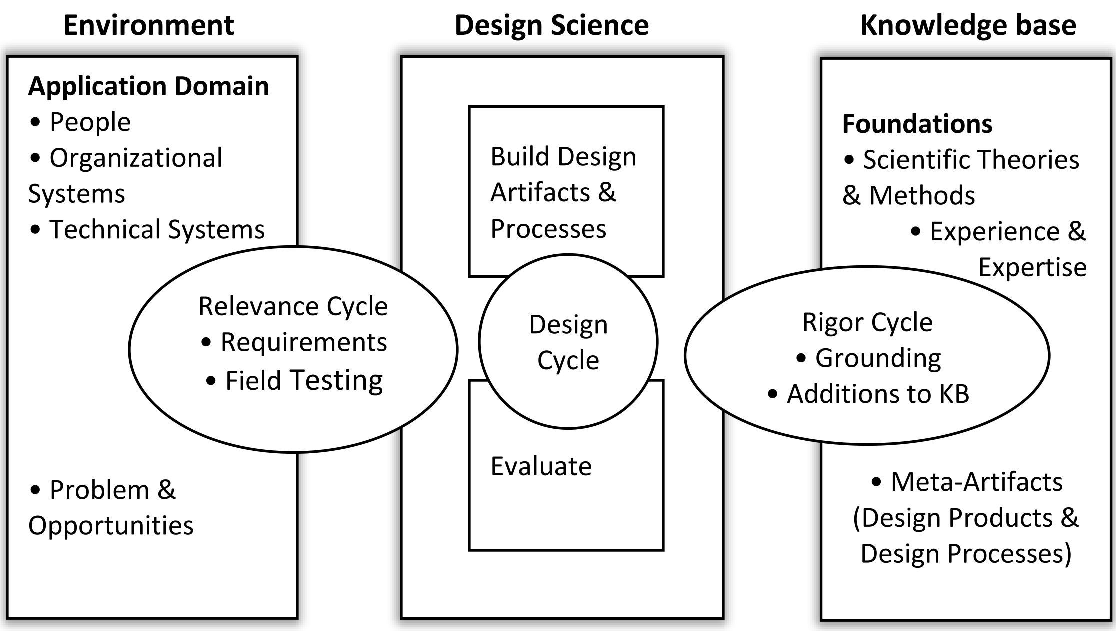


Figure 4: Design science research cycles, adapted from (Hevner A. R., 2007, p. 2)

The scientific debate on design science research is confusing. As a step towards providing clarity, Iivari (2015) has recognized two strategies used in design science research and explained the differences between them along 16 dimensions representing the context, outcomes, process and resource requirements (Iivari, 2015). Design science research produces conceptual or real system implementation (optional) meta-artifacts (Iivari, 2015). The primary goal of real time implementation is to evaluate the involved software and/or hardware. In the first strategy, discussed by the author (Iivari, 2015), researcher(s) construct or build IT meta-artifacts to address a class of problems. A specific client may or may not be involved while using this strategy to address the problems. In the second strategy, a specific client's problem is addressed by building concrete IT artifacts. This research work follows the second strategy mentioned by Iivari (2015) and the five steps process presented by Vaishnavi and Kuchler Jr. (2008) (see Figure 3). The next section explains the research steps used that follows the same five steps DSR approach. Different available IoT platforms are compared and one platform (IBM

Watson IoT for Bluemix) is evaluated by constructing digital artifacts (e.g. application). The IoT-A modelling approach (section 3.4) is followed to explain one IoT domain in detail.

2.2 Research Design Steps

Figure 5 represents the steps that are followed in this research study. The research begins with the introduction to the topic and explains the design methodology followed to design the research work. The awareness of the problem is studied through the literature analysis, and research objectives and research questions are identified. A tentative design is suggested by comparison of the different platforms by means of general, business and technical criteria. The existing platforms are compared to understand the functionalities they are offering. A use case is studied and implemented using an IoT development platform to further understand the capabilities of that platform. Finally, the research is concluded using the results obtained by the theoretical study and practical work.

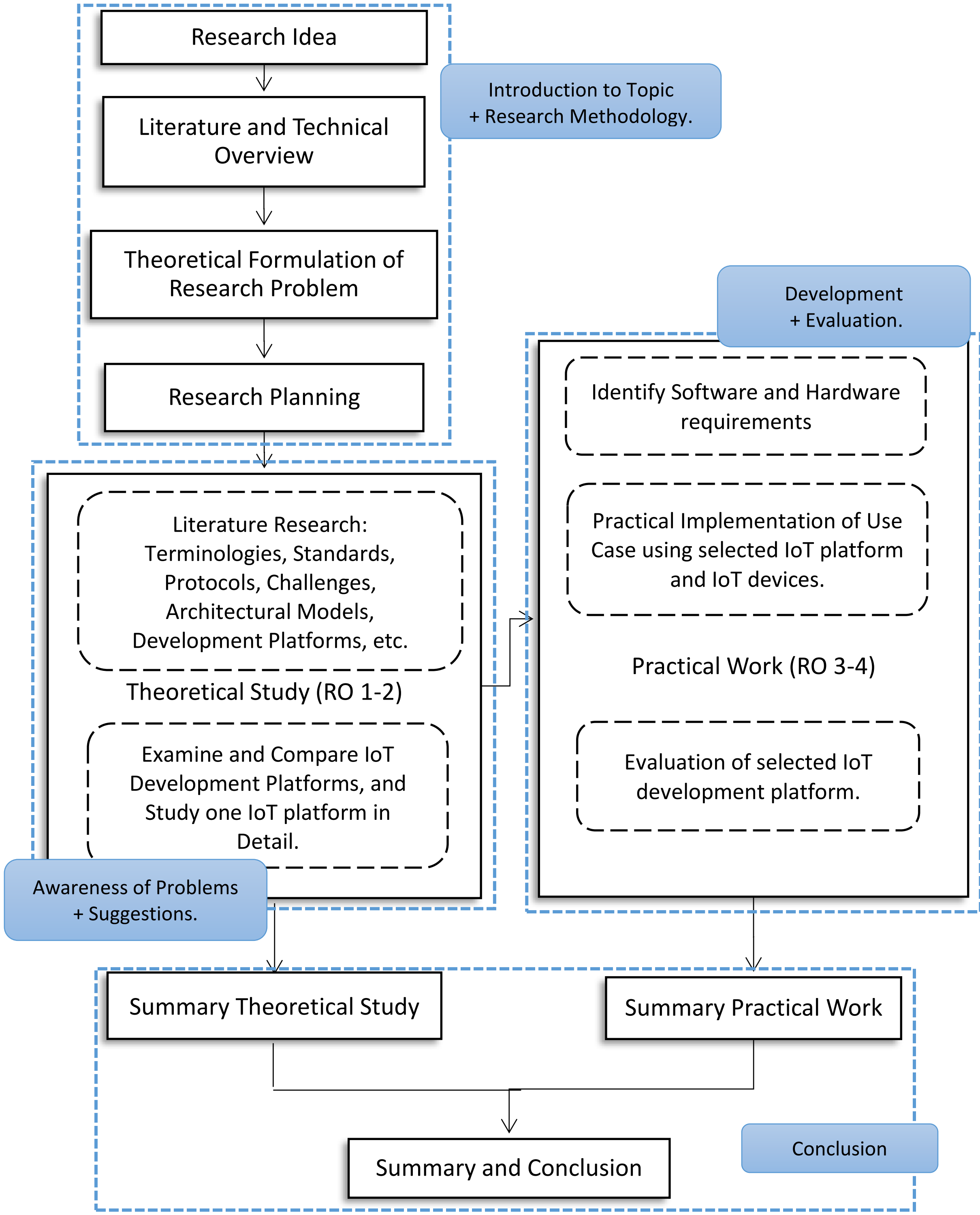


Figure 5: Research steps using DSR approach

Chapter 3. Theoretical Foundations

Following the design science research approach (see section **Error! Reference source not found.**), research problems have been identified through literature study. This chapter provides the theoretical foundation for the research work. The IoT domain and scope, ecosystem, challenges, modelling approach, different communication protocols, and cloud computing model have been discussed in this chapter.

3.1 Internet of Things (IoT) Domain and Scope

Internet of Things (or IoT), a term coined by Kevin Ashton in 1999 while searching for a way to link the idea of RFID in the Procter & Gamble (P&G) supply chain to the Internet (Ashton, 2009). The idea was to enable the computers to gather the data using the RFID and sensors technologies instead of manually entering it by a human being. The term IoT is interpreted in different ways since then, as there is no globally accepted definition of IoT. Today, in general, IoT refers to 'interconnect all things' (IEEE, 2015), that means, to connect the physical world (i.e. devices, TV, car, etc.) to the virtual technology platform (i.e. social media, clouds, etc.) with the help of the internet.

IoT has a wide scope in various research domains including radio frequency identification (RFID), machine-to-machine (M2M) communication and machine-type-communication (MTC), wireless sensor and actuator networks (WSAN), ubiquitous computing, web-of-things (WoT) etc. (Mazhelis, Luoma, & Warma, 2012). Multiple applications areas like automotive and machinery, home automation, consumer electronics, smart healthcare etc. will benefit from the IoT development.

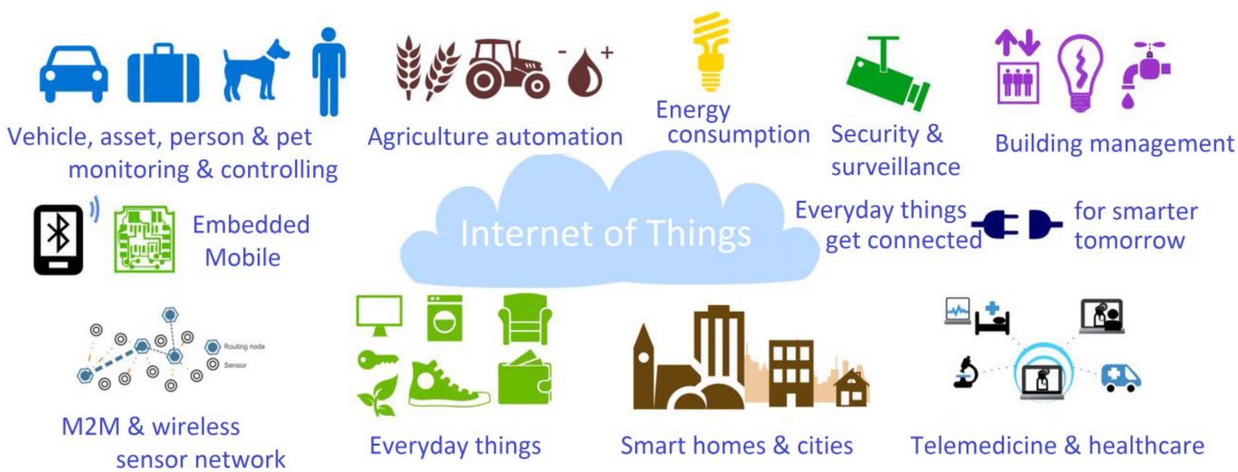
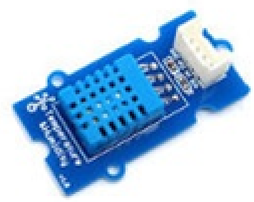


Figure 6: Internet of Things application areas, adapted from (Impe, 2014)

IoT can be realized as an umbrella term which unites related vision and underlying technology, and represents a convergence of multiple domains (Mazhelis, Luoma, & Warma, 2012). IoT is considered as a way to improve the quality of life for the society by interconnecting infrastructures, like power plants, traffic management, and safety information. The IoT vision cannot be possible without the sensors, actuators and IoT platforms interconnected through the Internet.

- A Sensor (see Figure 7) is a device that receives a physical quantity and converts it into a processable signal (e.g. optical, electrical, etc.). A wide range of sensors is available today, e.g. temperature sensors, humidity sensors, or light sensors, just to name a few.
- An active element of sensors is called Transducer, which converts one form of energy into another.

Grove – Digital Light sensor



Grove – Temperature and Humidity sensor

Grove – Air quality sensor

Actuators

Figure 7: Sensors adapted from (Williams, 2015) , and Actuators adapted from (Robot shop)

- In contrast to sensors, an actuator (see Figure 7) transforms an electrical signal into physical quantity. Actuators are used to control the devices. Some examples of actuators are linear actuators or rotatory actuators used to open/close windows.

The concept of IoT can be applied in different domains. This section explains the different possibilities where IoT can be effective. IoT can also be very useful in emergency and disaster management. When stricken by a disaster, we typically rely on the first responders and emergency managers (Govloop, 2015). Those managers, in turn, rely on the communication infrastructure to get the information about the affected area and to have proper coordination during the rescue operations. But, disasters like fire, earthquakes and floods also destroy the communication network. Without real-time information, it is difficult to evacuate people from the affected sites. IoT can be used to create an Ad-hoc communication network (Govloop, 2015). Connected sensors to affected areas can relay real-time information. One could also monitor the situation through videos and sensor information with the help of Unmanned Aerial vehicles (UAVs) and ground robots.

Using the sensors and cloud-based IoT platform (section 4.1), it is possible to implement various use cases in the different application area, for example, a cow farmhouse use case to monitor and control the barn climate and to track the movements of cows is discussed later in Chapter 5. The various studies show the current IoT progress in different domains. Below some of the ongoing projects (Govloop, 2015) are discussed that indicate the wide scope of IoT.

America Seattle smart buildings project: A group is formed by Microsoft and Seattle 2030 district for the America Seattle smart buildings project. The goal of this project is to meet certain energy reductions through real-time data analysis of Seattle buildings by 2030. The data is collected from the elevators, heating and cooling towers. The current program has four building with 547 sensors that are collecting 5000 data points in every three and a half minutes on average (Govloop, 2015). The data is stored in the cloud and accessible through a dashboard.

GSA data sensors: In a project at Washington by GSA (General Services Administration), GSA's data sensors, the goal is to connect the building's management system to the central cloud-based platform to improve efficiency. The idea is to have sensors everywhere. For example, a sensor can be used to detect how much sunlight shines through a window and indoor bulbs can be controlled to shade or dim accordingly; lights, air conditioners or power sources can be turned off when needed. Other sensors can relay information about utility consumption, such as how much cooled water or energy used in a day, and so on.

Allergen sensing: Through US Ignite, Chattanooga launched a pilot project GASP or Geo-located Allergen Sensing Platform. Sensors can be installed on public buses, or use as a wearable device. These sensors collect the data about the particulate matter in the air and report back the real-time data to GASP. A mobile web application has been created to analyse, interpret and display allergy-related information. The idea is similar to Google street view, to create a pollution view platform.

Nashville public transit (T-Hub, or Transit-Hub application): The T-Hub App tracks the bus location in real time, estimates trip times and receives alerts in different scenarios. A system that includes both hardware and software, uses a simulator to predict the traffic patterns and advanced analytics to predict the departure and arrival time for the buses. The city collects the sensors data from the buses every minute (or, as scheduled) and the information is posted to the server. All the data is updated every 10 seconds. T-Hub allows passengers to plan their trips and enables officials to assess and improve the routes and transit options.

A wide range of use cases and applications are available that shows how IoT is going to improve the lifestyle of society.

3.2 IoT Ecosystem

An ecosystem, in general, is a complex system of services, its elements and relationship among them (Chen, Hu, & Zhang, 2012). It's hard to define an IoT ecosystem due to the rapid expansion and vast possibilities in the IoT (IEEE, 2015). But, some work is done by different authors and standard bodies to define an IoT ecosystem. As mentioned by Mazhelis, Luoma and Warma (2012), an IoT ecosystem from business prospective can be defined as "the network of buyers, suppliers and makers of related product and services, plus the social-economic environment" (Mazhelis, Luoma, & Warma, 2012, p. 3). As there is a plethora of different IoT scenarios within different IoT domains, it is important to know the key players that are working together to shape the IoT future. As a part of the IEEE IoT initiative (IEEE, 2015), a study was conducted to define IoT ecosystem and answer questions in the context of IoT like what is missing from a business, technology and standardization point of view, what the technologies that enable the IoT growth are, and what the contribution of various bodies including researchers, industry and academic is, etc. The study discusses three principal areas of interest: market, technology and standard as shown in Figure 8. Providing a common standard for communication between the IoT devices and transfer of data between them is one of the big challenge ahead of IoT development process and is mentioned in section 3.3 along with other main IoT issues. The three key areas are discussed below in detail.

Market: IoT market is growing very fast in different segments. Many proprietary as well open source IoT solutions are already there in the market. A list of market players working in the direction to shape the IoT, as discussed in IEEE ecosystem study (IEEE, 2015), are given below.

- **Commercial players in the Offline world:** Produce "things" like smart appliances, home automation devices, personal devices (i.e. smartphones), smart automobiles, etc.
- **Commercial players in the Online world:** Provide services related to IoT, but some of them also produce "things". For example, IBM (Watson IoT for Bluemix), Microsoft (Azure IoT Suite) , Amazon (AWS), Google Glass (a "thing"), etc.
- **Research and Academia:** Create theories, educate a new generation of technologists, and business people to expand IoT.
- **Governments and Utilities:** Create the smart cities, the smart grids etc. The typical example includes traffic cameras, security cameras, adaptive traffic controllers, etc. Government also provides fund to shape the technologies.
- **Other players:** Entrepreneurs, consumers, regulatory authorities.

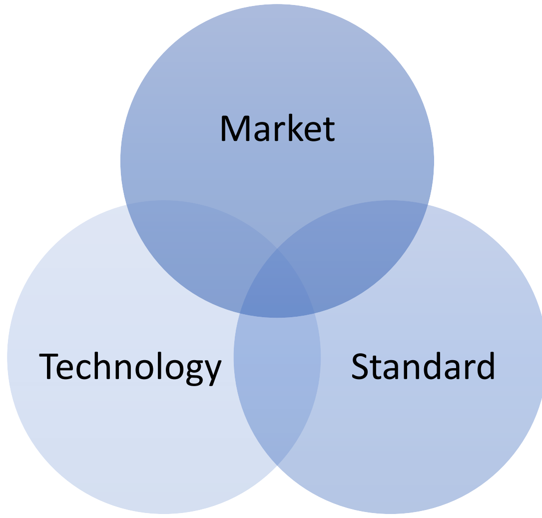


Figure 8: IoT principal areas of interest, as mentioned in (IEEE, 2015)

Technology: For the growth of IoT, it is essential to have the technologies that support this concept. Technology provides the infrastructure and platform for developing and deploying the IoT services and IoT applications. Various communication protocols are available which are adopted by the organisations to build their own IoT solutions. Some of the important protocols are discussed in section 3.6. The smart devices, sensors, actuators are now available at lower cost and more power efficient (IEEE, 2015). The various organisations have already come up with their cloud-based IoT solutions, some of them are analysed in detail in Chapter 4. The technology is evolving so are the challenges associated with it like interoperability or security which are discussed in section 3.3.

Standard: Standardization (also refer section 3.3) is necessary for a technology or concept to be implemented and accepted globally. Without the standardization issues like interoperability or security cannot be addressed properly. A standard for exchanging the data between the devices, communication protocols, specifications for the smart devices etc. is needed for the global success of IoT and to achieve the ITU-T vision of 'anything connected'. There are many standard bodies that are working in defining the concept of IoT. IoT-A has proposed an IoT architectural reference model (IoT-A, 2013) in addition to defining the key building blocks. IEEE is also working to develop an architectural framework for IoT to promote cross-domain interaction, functional compatibility, and to aid system interoperability (IEEE, 2016). ETSI end-to-end architecture has three IoT technical domains including device, connectivity, and service. IoT provides the business opportunity for telecom operators, application and service providers, platform providers and integrators. This can only be possible by having proper standards in these domains. Some of the important standard bodies working in the field of IoT are following (IEEE, 2015).

- Institute of Electrical and Electronics Engineers (IEEE): www.ieee.org
- International Electrotechnical Commission (IEC): www.iec.ch
- International Organization of Standardization (ISO): www.ios.org
- International Society of Automation (ISA): www.isa.org

- International Telecommunication Union (ITU): www.itu.int
- Internet Engineering Task Force (IETF): www.ietf.org
- World Wide Web Consortium (W3C): http://www.w3.org

3.3 IoT Challenges

The IoT brings many possibilities in various domains as discussed in the previous sections, but, significant challenges persist. In this section, key emerging technical and social challenges related to IoT are presented. As IoT success is only possible through global acceptance of IoT standards and IoT policies, the role of private vendors and government cannot be ignored.

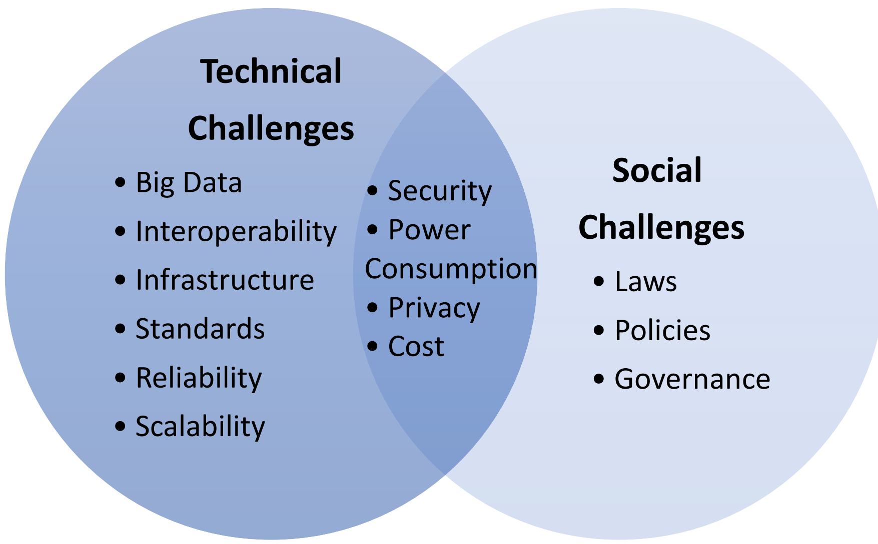


Figure 9: IoT technical and social challenges, own illustration

Some of the IoT challenges (IEEE, 2015) (Al-Fuqaha, Guizani, & Mohammadi, 2015) (Govloop, 2015) (Rose, Eldridge, & Chapin, 2015) as shown in Figure 9 are discussed below in detail.

- **Big-Data:** A large amount of real-time data will be produced by IoT devices as the number of connected devices grow with time. To handle and store this Big-data, and get the insight from this huge volume of data, is a big challenge in front of IoT solution providers (IEEE, 2015), as this huge amount of data will exceed the capacity of commonly used hardware and software tools (Al-Fuqaha, Guizani, & Mohammadi, 2015). Also, there are questions of monetising big data (IEEE, 2015).

- **Interoperability:** The different IoT solution providers are using open or proprietary solutions to enable IoT devices, applications, and IoT platforms to communicate with each other. Some of the protocols that can be used for IoT development are discussed in section 3.6. To provide a universal solution, it is required that the devices, applications, and platforms are interoperable.

- **Security:** Connecting physical devices (i.e. home appliances, cars, etc.) to the Internet will make them vulnerable to the Internet security threats present in the Cyber-World (Govloop, 2015). The IoT devices which are not supported by the device vendors anymore (e.g. for which no device update is available) are more likely to be a security risk (Govloop, 2015). To continuously provide the security updates and make the users aware of the current or possible threats is going to be very challenging for the IoT solution providers, as, without it, the devices will be more vulnerable to the threats like cyber-attack (Govloop, 2015).

- **Privacy:** Privacy is a serious concern in this digital world. The issues related to ownership of data, authorization to act based on the data, etc. have both technical and social aspects attached with it (Govloop, 2015). Today, organisations especially in retail, manufacturing, and service provider sectors are collecting and using the personal information of their users for improving their services (e.g. profile based recommendation) so as to increase business gains. It is challenging to take users in confidence while privacy of their data is a concern.

- **Laws and Policies:** The IoT has a global presence. Therefore, the laws and policies should be globally accepted. Also, in many countries, laws protect to use the confidentiality and security of information in different sectors (e.g. Healthcare information) (Govloop, 2015). The lack of access to the data makes it difficult to provide all necessary information needed for an IoT enabled device to work properly. Therefore, such privacy issues need to be addressed properly by having laws and policies that help the technological innovation rather than create a barrier for it.

- **Infrastructure:** The challenge is to construct the communication infrastructure using interoperable devices (Usländer, 2014). As Internet plays an important role in the success of IoT, the issues related to the communication network (i.e. unavailability of the network, fault tolerance, etc.) should be addressed adequately.

- **Standards:** The requirement of IoT standards (e.g. communication, protocol, storage, security, etc.) should be made a priority in order to handle the challenges discussed above. Standardization provides interoperability, compatibility, reliability and effective operations at a global level (IEEE, 2015). There are many standard bodies (see section 3.2) working in the direction to define IoT and related terms, but, still, a joint effort is needed.

- **Reliability and Scalability:** A reliable and good quality of service in different conditions (i.e. emergency situations, bad weather etc.) is also important (Al-Fuqaha, Guizani, & Mohammadi, 2015). A good Information and Communication (ICT) infrastructure plays an important role in providing reliable services. The offered IoT system should also be scalable to match the growth of IoT.

- **Power Consumption:** The battery life, consumption of power during the span of service has both technical and social challenges associated with it. The energy consumption is a concern for the subscribers, while uninterrupted service is a priority of the service provider (Rose, Eldridge, & Chapin, 2015).

- **Cost:** The cost of IoT service, different related devices (sensors, actuators, etc.) in addition to the reliability has a big impact on the business. To provide a low-cost solution, flexible pricing is very critical in this highly competitive environment (Rose, Eldridge, & Chapin, 2015).

- **Governance:** To provide a quick and global IoT solution, legal issues should not make the things complicated. The governments should provide flexible rules, approval of ideas, certification availability to make things smooth for the emerging technologies (Sundmaeker, Guillemin, Friess, & Woelfflé, 2010).

The various technical challenges presented above are discussed in this report in the context of existing cloud-based IoT development platforms. The social aspects (laws, policies, governance) are not covered in this research study any further.

3.4 IoT-A Modelling Approach

IoT-A has introduced an IoT Architectural Reference Model (ARM) (IoT-A, 2013) which provides a common language that everyone involved in the IoT development process can understand. Various IoT platforms are using different protocols, data transfer formats, therefore, interoperability of IoT solutions need to be ensured at the communication as well as the service level across these platforms. IoT-A ARM provides a way to design an IoT solution that is easy to understand and independent of any specific language. The various models introduced in the IoT-A ARM are explained below. A more concrete example is shown in section 5.2 that uses some of the following modelling approaches.

- **Domain model**: Domain model, the primary and key model, provides a top-level description of the domain (IoT-A, 2013). It describes the generic components of a generic architecture. The main concepts like devices, IoT services and relationship between them are represented using this modelling approach. It provides the common vocabulary of the IoT domain. It also helps in structuring an application scenario. To graphically illustrate the model, UML (Unified Modelling Language) approach is followed. The following colour schemes are used to depict the concepts. The box with the blue colour represents hardware, the green colour represents software, yellow colour represents animate beings (humans, animals etc.), and multiple or no category is represented by the brown colour (IoT-A, 2013). The colour scheme is also followed while creating a domain model representation (Figure 31) for the use case, as discussed later in this research report.

- **Information Model:** The information model explains the way to model IoT knowledge which is based on the IoT Domain model. The structure (e.g. relations, attributes, services) of IoT-related information is defined at a conceptual level. The information explains the answers to questions like who, what, where and when. The information model shows the common concepts from the Domain Model, for example, virtual entities and services.

- **Communication Model:** It explains communication between IoT devices and the Internet as a whole. It identifies homogeneous subsystem and their capabilities and identifies the suitable protocol stacks and network topologies to be merged in a common system view. It also defines the gateways and other bridging solutions.

- **Functional model**: The functional model identifies the groups of functionalities. For example, a key functionality in any distributed system is the communication between its different components.

- **Trust, Security and Privacy model**: For any system trust, security and privacy are very important. Such aspects can be explained by following this modelling approach.

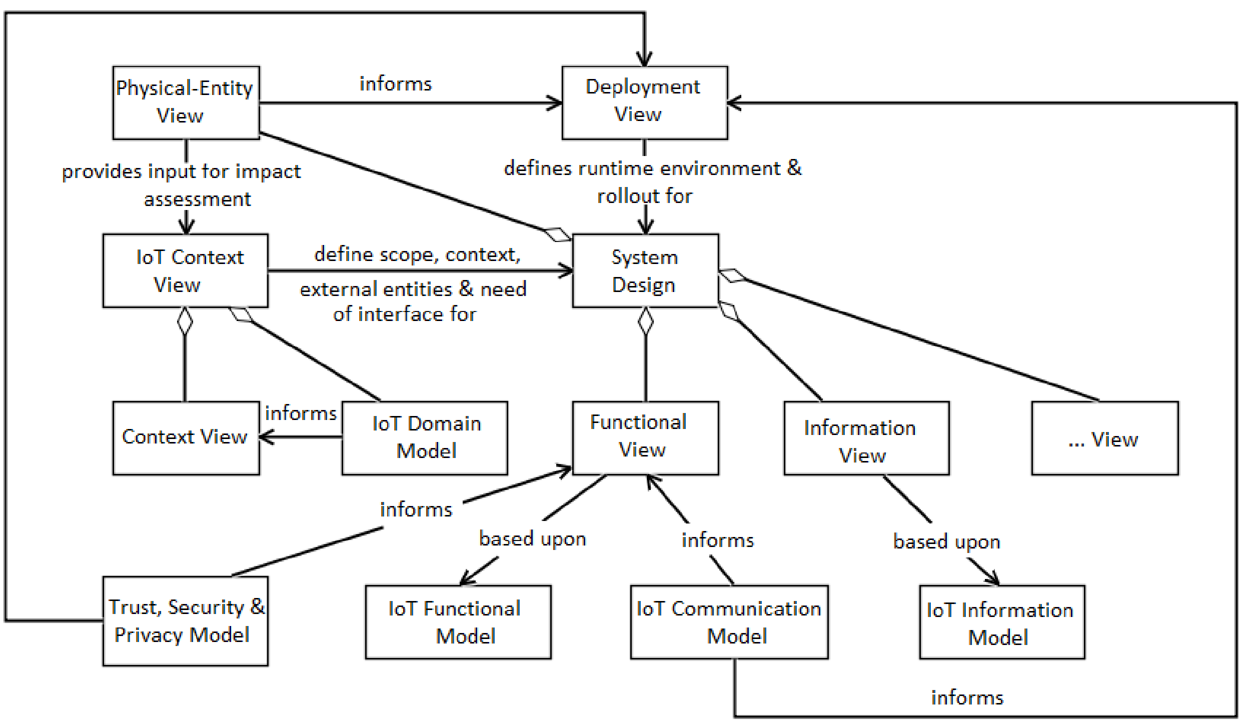


Figure 10: Architectural view relationships, adapted from (IoT-A, 2013, p. 162)

IoT-A ARM also explains the architectural view including physical view, context view, functional view, information view, and deployment view. The relationship between different views and models is shown in Figure 10.

3.5 Cloud Computing

The term cloud, along with describing a technology, is also used to describe a way to model business offerings. Cloud computing is a delivery of computing as a service, where a pool of physical and/or virtual resources are being shared, rather than deploying a local or personal hardware and software (Marechaux, 2015). There are three service levels of cloud computing (Amazon, n.d.-a):

- **SaaS** (Software as a Service) provides business services to customers. This is a complete product that is run and managed by the service provider. Examples: Web-based email.
- **PaaS** (Platform as a Service) provides the platform where you can build and deliver cloud applications. Examples: IBM Bluemix, Microsoft Azure, etc.
- **IaaS** (Infrastructure as a Service) is the basic building block for cloud technology and provides the hardware, storage, and network capabilities. Example: Amazon Web Service (AWS).

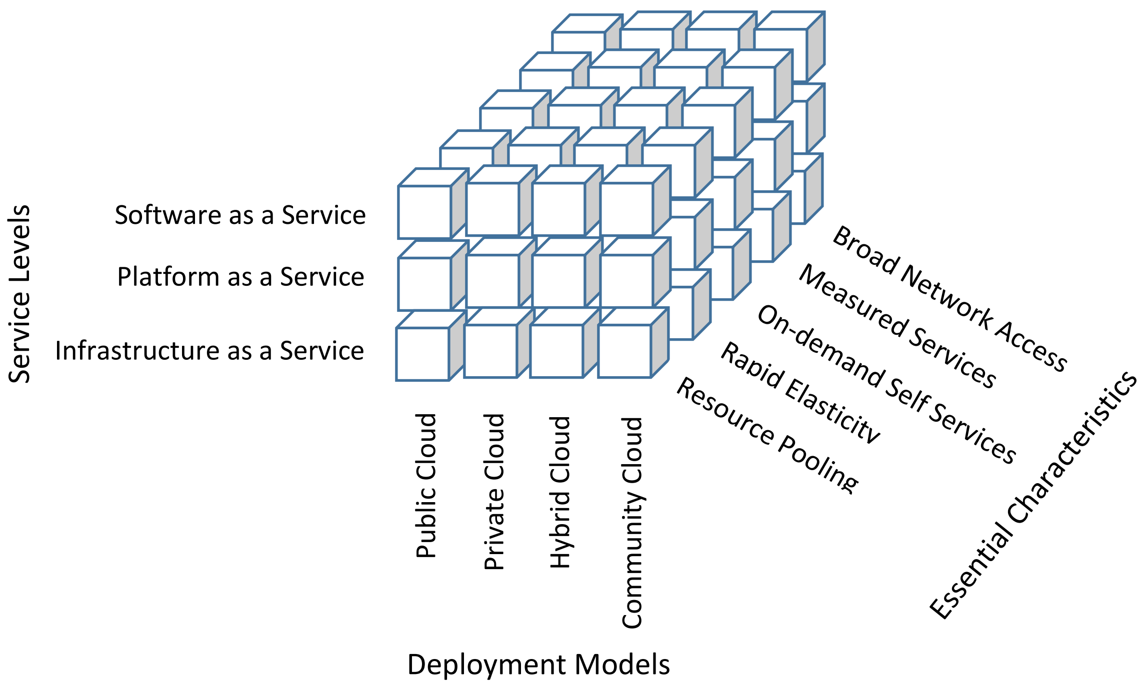


Figure 11: Cloud computing model, adapted from (Craig-Wood, 2010)

According to NIST (National Institute of Standards and Technology) definition of cloud-computing (Mell & Grance, 2011), cloud model (see Figure 11) consists of five essential characteristics (broad network access, measured services, on-demand self-services, rapid elasticity, resource pooling), three service levels (software as a service, platform as a service, infrastructure as a service) and four deployment models (public cloud, private cloud, hybrid cloud, community cloud). The broad network access capability provides the cloud access through mobile phones, laptops and PDA's. The computing resources (e.g. storage, processing, memory, network bandwidth, and virtual machines) should be provided according to consumer's need. The public cloud is accessible to the general public and owned and maintained by independent vendors. The private cloud is only available within an organisation for its internal users. The community cloud may exist on premise or off premise is shared by several organisations. The hybrid cloud is a combination of private, public or community clouds. The public cloud is commonly referred to Cloud Computing. The different IoT platform discussed in section 4.1 are cloud-based platforms. Cloud-based platforms for Internet of Things available today provides functionalities to send and receive data from the IoT devices. In addition to this, they also provide data analytics, data visualization etc. In this report, we have analysed different Cloud-based Internet of Things platforms (section 4.1).

3.6 IoT Communication Protocols

For the communication in the Internet and other computer networks, the Internet Protocol Stack is used which consists of four layers: Application, Transport, IP and Link layer. TCP (Transmission Control Protocol), a transport layer protocol, provides end-to-end communication with reliable, error-free, and in-sequence delivery of IP packets. TCP uses a three-way handshake to guarantee the reliable connection between endpoints. UDP (User Datagram Protocol) is less reliable, and delivery and duplication are not guaranteed. REST (Representational State Transfer), an architecture style, is used to design networked applications (Elkstein, 2008). RESTful applications use HTTP requests to create/read/update/delete data. In this section, we have discussed the various communication protocols that can be used while transferring data between IoT devices and Cloud platforms, or Cloud platform and IoT applications.

3.6.1 HTTP/HTTPS

HTTP, IETF Standard (HTTP/1.1 in RFC 2616), is an application layer protocol that operates over the TCP (or similar) protocol and is used to transmit data in the Web. Transport Layer Security (TLS) and its predecessor Secure Sockets Layer (SSL) are used for encryption and decryption of packets of data when it moves from one end point to another. This information hiding is done to prevent Eavesdropping as unsecured web communication is vulnerable to cyber-attacks. HTTPS is a secure HTTP protocol that uses the SSL/TLS framework. HTTP/HTTPS uses the request/response approach (see Figure 12).

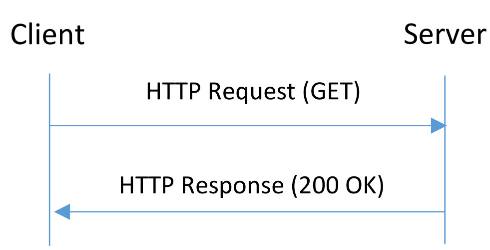


Figure 12: HTTP request/response approach, adapted from (Fielding, et al., 1999)

By default, HTTP/HTTPS is a stateless protocol which means that it does not maintain the state of the connection and the connection terminates after each response to the requested query.

For the new query, a new connection is established which makes it difficult to use HTTP alone (Karasiewicz, 2013) in IoT scenario as IoT devices are generating a large volume of data in real-time and connecting/disconnecting for every query will add more work for the server. Therefore, a need of light-weight reliable protocol arises to handle communication between devices and clouds in IoT environment. Some of the available light-weight protocol (e.g. MQTT) are discussed below.

3.6.2 WebSocket

The WebSocket (Kaazing Corporation, n.d.) is a stateful, full-duplex, and always connected architecture. The Websocket, designed as a transport layer protocol, is a complement to HTTP. The communication is made through a single socket between client and server. Other protocols such as MQTT (Message Queuing Telemetry Transport), AMQP (Advanced Message Queuing Protocol), pub/sub can be run on top of WebSockets to provide more flexibility and power. HTML5 WebSocket is a standard that helps in building scalable real-time web applications. WebSocket supports to send both text and binary frames at the same time in either direction.

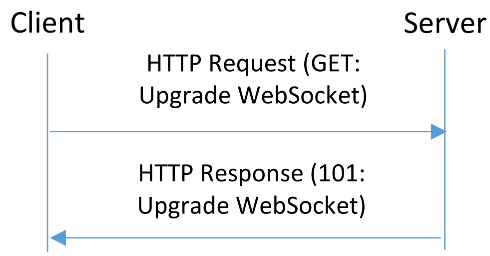


Figure 13: WebSocket handshake, as mentioned in (Kaazing Corporation, n.d.)

The connection between client and server remains open once the connection is upgraded to WebSocket from HTTP protocol during their initial handshake (Figure 13) and, therefore, there is no need to send another request to the server as was the case with HTTP.

3.6.3 CoAP

CoAP (Constrained Application Protocol), an IETF standard (IETF, 2014), is a web transfer protocol for resource constrained nodes and constrained (e.g., low-power, lossy) networks. The constrained nodes often have a small amount of ROM and RAM. CoAP protocol is designed for M2M applications (i.e., smart energy and building automation). CoAP optimises a subset of HTTP functions for constrained environment (Sheng, et al., 2013). This protocol is based on request/response model and uses datagram-oriented protocol (i.e. UDP).

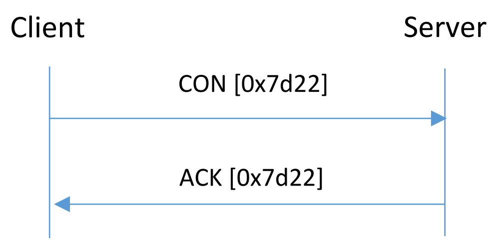


Figure 14: CoAP - Reliable message transmission, as mentioned in (IETF, 2014)

Reliability is provided by CON (confirmable message) and its ACK (acknowledgement) with the same message ID (Figure 14). It also has built-in resource discovery.

3.6.4 AMQP

AMQP (Advanced Message Queuing Protocol), an OASIS and ISO 19464 standard (1.0), is an open messaging protocol (OASIS, 2012) that enables to build cross-platform apps using different vendor's broker, libraries and framework. The main features are reliable communication (exactly-once delivery) and interoperability. It was designed as an open replacement for existing proprietary messaging middleware.

Figure 15: AMQP communication model

Figure 15 shows the main components of AMQP communication model. The client publishes the message to the Exchange (a part of broker) where it is processed and routed to the queue. From the queue, the message is sent to the Client who has subscribed for this. Binding module routes the message between Exchange and Queue (by default binding maps routing-key to Queue name). AMQP connection provides full-duplex, reliably ordered sequence of frames. Microsoft's IoT solution supports multiple protocols such as AMQP, AMQP over WebSocket, and MQTT (section 3.6.5).

3.6.5 MQTT

MQTT (Message Queuing Telemetry Transport), an OASIS standard, was originally developed by Andy Stanford-Clark (IBM) and Arlen Nipper (Eurotech) in 1999 for monitoring an oil pipeline through dessert (HiveMQ). The goal was to develop a bandwidth-efficient and battery efficient protocol. MQTTv3.1, suitable for IoT communication, is a lightweight protocol that is based on publish/subscribe model. MQTT also supports QoS (Quality of Service) control.

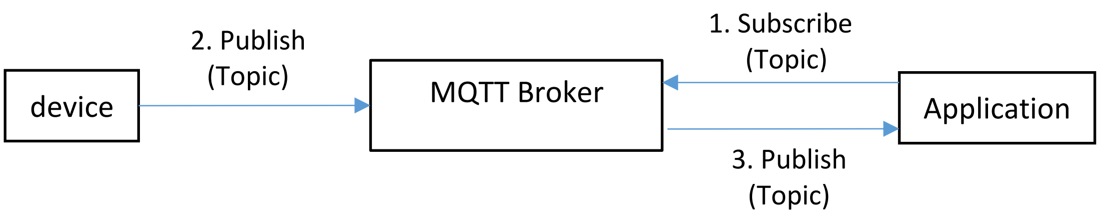


Figure 16: MQTT publish/subscribe architecture, adapted from (HiveMQ)

MQTT is event-driven and enables the messages to be pushed to a subscriber. The subscriber, subscribed to some topic, receives the published value from the IoT devices corresponding to that topic through the MQTT broker. A topic can be a simple string that can have hierarchical levels.

3.6.6 Protocols Comparison

	HTTP/HTTPS	WebSocket	CoAP	AMQP	MQTT
Standard	IETF	IETF	IETF	OASIS & ISO	OASIS
Messaging	Request/ Response	Request/ Response	Request/ Response	Publish/ Subscribe	Publish/ Subscribe
Transport	TCP	TCP	UDP	TCP	TCP
Security	HTTP is not secure. HTTPS, runs over SSL/TLS is secure.	WebSocket (ws://) is not secure. WebSocket over SSL/TLS (wss://) is secure.	CoAP uses DTLS (Datagram Transport Layer Security).	AMQP uses TLS or SASL (Simple Authentication and Security Layer).	By default, MQTT does not use any encryption. But, "secure-MQTT" with SSL/TLS security can be used.
Quality of service (QoS)	No QoS	No QoS	Yes, available	Yes, available	Yes, available

Table 1: IoT protocols comparison

The lightweight IoT protocols such as CoAP, AMQP, MQTT are preferred for device to device and device to cloud communication. HTTP/HTTPS and WebSocket are more suitable for web to web communication (e.g. communication between cloud and IoT application). The comparison of different IoT communication protocols is discussed in Table 1. All the communication protocols discussed above are used by different IoT service providers depending on the requirement of their IoT solution. Some of the IoT development platforms support multiple protocols (see section 4.2). In the next chapter, the different cloud-based IoT platforms that make use of the above-discussed protocol(s) to deliver the IoT solution are discussed.

Chapter 4. IoT Development Platforms

The *Suggestion* in a design sciene research is a creative step where existing capabilities of available systems are visualized to gain knowledge and provide theoretical base for the problem area. The design science process step (*Suggestion*) is followed to compare the existing IoT platforms. A number of open source and proprietary IoT development platforms are available in the market. For this research study, the popular emerging IoT development platforms from different key vendors are presented and then compared in terms of their capabilities. In this report, the IoT solutions from IBM, PTC, Microsoft, Google, and Amazon are compared. This chapter starts with a brief introduction to the respective organisations and their IoT solutions including IoT development platforms.

4.1 Vendors and their IoT Solution

This section explains the various existing vendors and their work in IoT development.

4.1.1 IBM: Watson IoT for Bluemix

This section gives a brief introduction to IBM's IoT approach.

A Brief Company Profile

IBM (http://www.ibm.com/us-en/)	
A brief introduction	IBM (International Business Machines Corporation), founded on 16th of June, 1911, is a technology company. It operates in five segments: Global Technology Services (GTS), Global Business Services (GBS), Software, System Hardware and Global Financing (Reuters, n.d.-a). Softlayer, IBM's worldwide Infrastructure as a Service (IaaS) offering, is a single platform for public cloud servers, private clouds, and bare metal servers. IBM Bluemix, company's open cloud development platform, is a Platform as a Service (PaaS) offering that works over the Softlayer. IBM Bluemix provides mobile and web developers access to IBM software for integration, security, transaction, etc. The service catalogue available on Bluemix includes DevOps (from Development to Deployment), big data, mobile, Watson (for building cognitive applications), business analytics, data management, web and applications, security, Internet of Things, and integration. Watson Internet of Things (IoT) for Bluemix lets an application to communicate with the connected devices and consumes data received from those devices.

IBM (http://www.ibm.com/us-en/)	
IoT platform	Watson IoT for Bluemix.
Industrial partners in IoT	IBM Watson Internet of Things partners (IBM, n.d.-a): **IoT devices:** Arrow, Avnet, Flex, Ionics, element14. **Silicon and sensors:** Raspberry Pi, Bitreactive, Nexcom, ADLink technology, STMicroelectronics, Arduino, Intel, Texas Instruments, ARM, Mediatek, u-blox. **Gateways:** B&B Electronics, Elecsys, Intel, Multi-Tech Systems, Cisco, Plat'Home, Semtech. **Networks:** AT&T, Sprint, Jasper. **Cloud solutions:** Weather company, ESRI, FlowThings. **System integrators:** IBM Global Business Services, HCL, Jena, Sogeti Hi-tech/Cap Gemini.
Awards or nominations	• Postscapes IoT awards: Nominated in best technical enabler 2014 (Postscapes, 2014-a). • Ventana research award: IBM Bluemix for Cloud Computing won the technology innovation award (Ventana Research, 2015). • Inquirer Tech Hero awards: IBM IoT Foundation won best IoT product - 2015 (Inquirer, 2015).

Table 2: IBM - A brief profile

Watson IoT for Bluemix

IBM Bluemix, IBM's Platform as a Service (PaaS) offering, runs over Softlayer which is IBM's worldwide Infrastructure as a Service (IaaS). IBM Bluemix also provides the hosting capability.

The Watson IoT for Bluemix enables your application to communicate with the devices, gateways etc. MQTT (Message Queue Telemetry Protocol), a lightweight messaging protocol that uses Publish/Subscribe architecture, is used to communicate with the devices and gateways. As shown in Figure 17, using REST and real-time applications, sensors data can be pushed into the application.

The device needs to be registered with the IBM Bluemix before creating MQTT connection with the Watson IoT MQTT Broker. Watson IoT Service receives the data coming from the device

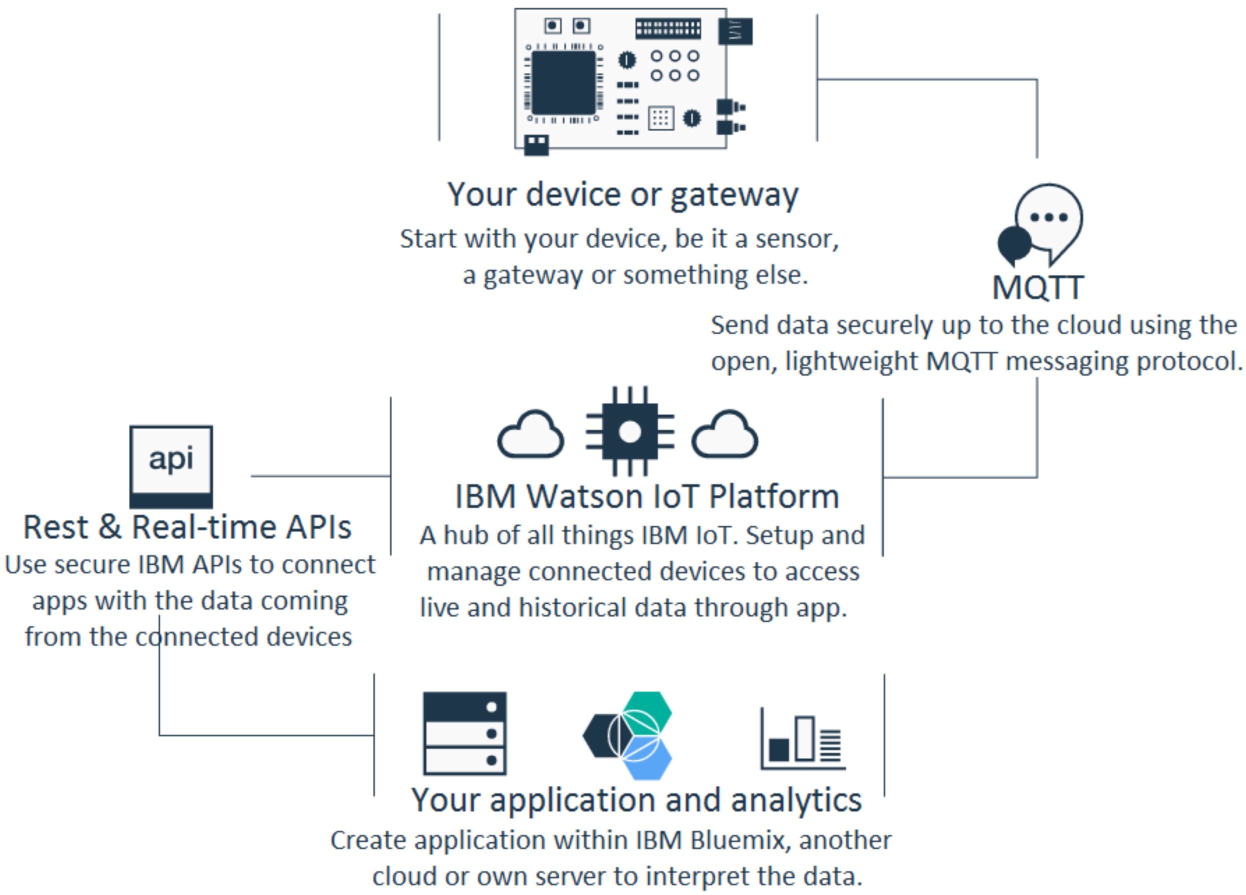


Figure 17: Watson IoT for Bluemix overview, adapted from (IBM, n.d.-b)

after successful MQTT connection and sends it to the application created on the Bluemix platform. Rules to handle the device data can be defined using the Node.js runtime application hosted on the Bluemix platform. IBM Bluemix supports runtimes in many languages including Java, Node.js, Go, PHP, Python, Ruby, and Swift. Bluemix also provides support for data analytics, storage, business analytics, etc.

4.1.2 PTC: ThingWorx

This section gives a brief introduction to PTC's IoT approach.

A Brief Company Profile

PTC (http://www.ptc.com/)	
A brief introduction	PTC (Formerly Parametric Technology Corporation), founded in 1985 (PTC), develops and deploys software and service technology solutions. In 2014, PTC acquired ThingWorx (a company founded in 2009) that placed PTC as an emerging player in the Internet of Things business. In the same year 2014, the acquisition of Axeda by PTC accelerates the IoT value. The ThingWorx platform enables rapid development and deployment of IoT solutions. Managing connected products and machines is made possible by the use of Axeda's cloud-based service and software solution.
IoT platform	ThingWorx.
Industrial partners in IoT	List of ThingWorx partners (ThingWorx, n.d.-a): **IoT solution provider:** Aquamatix, BioSerenity, Bosch Software Innovations, Camgian Microsystem, Dattus, DEPsys, DeviceLynk, Devicify, DVM IoT Solutions, Elisa Corporation, Gemini Design, Icuro, IoT Solutions, iQuest, Kepware Technologies, Mindtree, Mtuity, Nippon Systemware, OnFarm, Pactera, Reply, SmartPatch, Smoove Bike, Trimble, Vantiq. **Communication service provider**: AT&T, Bell, Cisco Jasper, CLX, FYBR, Novatel Wireless, Tele2 M2M, Telkomsel, Verizon. **Device cloud providers:** Stream, wot-io. **Edge and embedded providers:** Analog Devices, ARM, ATrack, BB SmartWorx, Broadcom, CalAmp, Cimetrics, Cisco Systems, Diamond Technologies, Encore Networks, FYBR, Gemalto, Grid Connect, iBot Controls, Ineda Systems, Intel, Kontron, Lanner Electronics, Lantronix, Libelium, Link Labs, Microchip Technology, Multi-tech Systems, NetComm Wireless, NXP Semiconductors, Option, Pointer, Qualcomm, SMC, Systech, Texas Instruments, Vantron, Velio, Wind River, Wi-NEXT.

PTC (http://www.ptc.com/)	
Industrial partners in IoT	**Independent software vendors:** Analytika, Cryptosoft, Dakota Fluid Power, Dat-uh, Device Authority, Digibe Software, Eurotech, Falkonry, Glassbeam, Gravity, Infobright, Initial State, Kore, LumenData, Mtell, Nurego, ParStream, Predixion, Real Time Innovations, Salesforce, ServiceMax, Splunk, SQLstream, Support, TruePosition, ViziApps, WiseUp Building Analytics. **System integrators & vars:** Acumen Solutions, Altimetrik, Architech, Aricent, Autoware, Axcend Technologies, Bilot, BM Communications, Callisto Integration, Celfocus, Cognizant, CSC, Dedicated Computing, Dell, doubleslash, DVM IoT Solutions, EPAM Systems, etechnologies group, exceet Secure Solutions, Fresh Gravity, Genpact, Geometric Limited, Giza Systems, Hitachi High-Tech, Ideology, IGATE, INDEFF Group, Infosys, Innovative Digital Experience, Interstates Control Systems, InVMA, IoT Devzone, IoT Solutions, ITC Infotech, Kalypso, KPIT Technologies, LT Technology Services, Lantern Technologies, Leverage, Mobisec Technologies, NETGENIQ, New Vision Engineering, Nexcom, Now Business Intelligence, Oberon Technologies, Percall, ProCom, Redsalt, Remote Grid, Software Factory GMBH, Spark Digital, Swift Solutions, Tata Consultancy Services, Tech Mahindra, TeraCode, Thaumatec, Thought Solutions, Transition Technology, UbiQuai, Venetia Systems, Venture Vertex, Volansys, Weidenhammer, Wipro, Zenith.
Awards or nominations	• Postscapes IoT awards: Nominated in best IoT technical enabler 2013. • Postscapes IoT awards: Winner of 'Technical enabler: Platforms and tools', and 'Must-follow IoT company' (Postscapes, 2014-b). • CES awards: Winner of enabling technology of the year and App development/enablement platform of the year 2015 (PTC, 2015).

Table 3: PTC - A brief profile

PTC ThingWorx

ThingWorx is a model-based application development platform that provides a complete application design, runtime, and intelligent environment. PTC Axeda applications are fully integrated into the ThingWorx and support remote monitoring, remote service, and remote access. The development effort using ThingWorx platform is faster due to the model-based design and search-based intelligence. The core application development features are the following (ThingWorx, n.d.-b):

- **ThingWorx Composer™:** It is an end-to-end application modelling environment. With the help of Composer, it's very easy to model the "things", business logic, visualization, data storage, security, and collaboration.

- **ThingWorx 'Drag and Drop' Mashup builder:** A ThingWorx web page is known as Mashup. Visualization of web pages can be created using the Mashup builder tool.

- **ThingWorx SQUEAL™ (search, query, and analysis):** SQUEAL is an advanced search feature that allows searching through all data in a model. The search can be done on the basis of entity names, model tags, data tags, wiki and blog content.

- **ThingWorx event-driven execution engine** and **three-dimension storage:** ThingWorx platform's event-driven execution, and three-dimensional storage that can store, relate, and expose the massive amount of data from people, systems, and connected "things" – makes the data valuable and actionable.

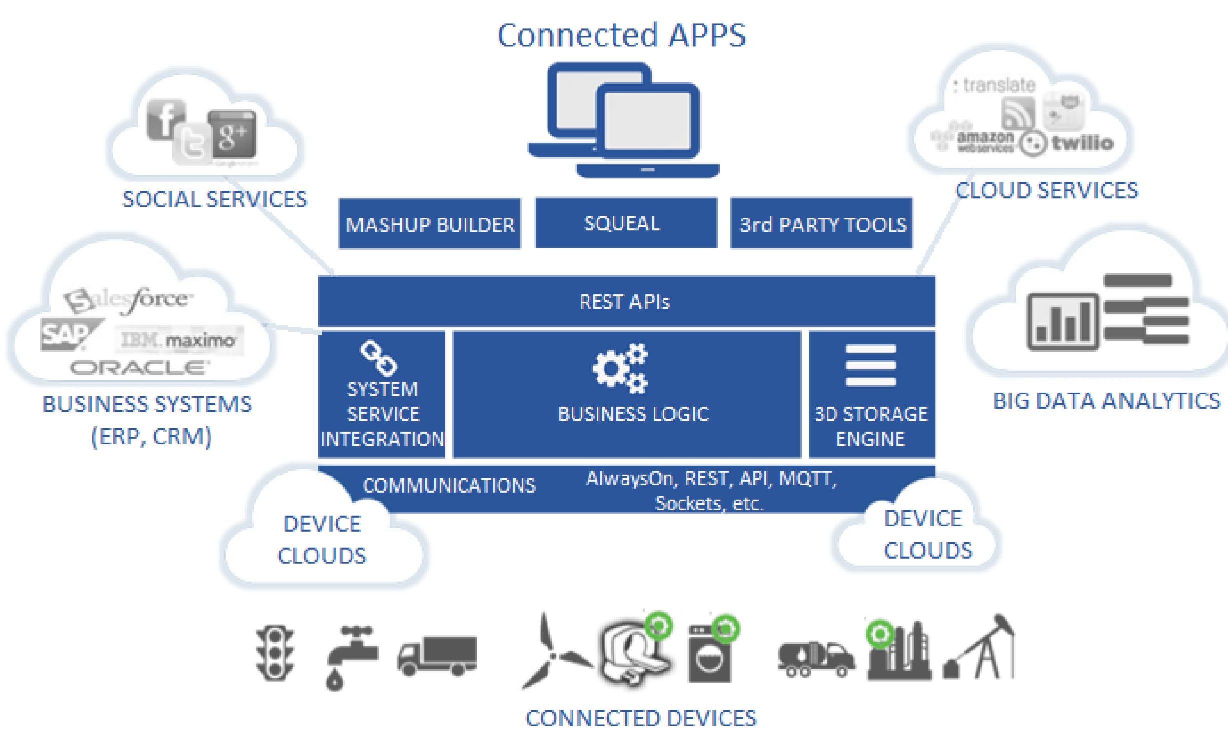


Figure 18: ThingWorx IoT platform, adapted from (Bullotta, 2011)

- ThingWorx connection with devices is very flexible and supported by several methods: Third party device clouds, Open APIs, AlwaysOn™ connectivity using Edge MicroServer of ThingWorx, and direct network connections.

4.1.3 Google: Cloud Platform for IoT

This section gives a brief introduction to Google's IoT approach.

A Brief Company Profile

Google (https://www.google.com)	
A brief introduction	Google, founded in 1998, is a multinational corporation deal in internet-related services and products. Google's Cloud Platform offers many products and services including compute, storage and databases, networking, big data, machine learning, management tools, developer's tools, and identity and security (Google, n.d.-a). Google's cloud platform for IoT supports storage, processing and analysis of data generated by smart devices. The data is received through the Ingestion module (Figure 19) that consists of cloud Pub/Sub (real-time messaging service), cloud monitoring and cloud logging. In early 2014, Google acquired Nest Labs, a company that works in delivering smart-self-learning thermostats, smoke detectors, and other security systems. Google has also launched Brillo, an operating system for the Internet of Things. Brillo has three major components: embedded OS (based on android), core services, and developer kit. The core services include Weave, a communications platform for IoT devices.
IoT platform	Google cloud platform for IoT.
Industrial partners in IoT	Intel, Marvell, Qualcomm, NXP, Imagination are partnered with Google to create boards for Brillo (Google, n.d.-b).
Awards or nominations	• Postscapes IoT awards: Nest Lab (pre-Google) was nominated in breakout start-up of the year 2013. • Lifehacker awards: Google won the best cloud computing provider award (Reader's choice) 2014 (Kidman, 2014).

Table 4: Google - A brief profile

Google Cloud Platform for IoT

Google cloud platform with the combination of other elements is used to provide an end-to-end IoT solution. As shown in Figure 19, IoT data can be managed at various stages:

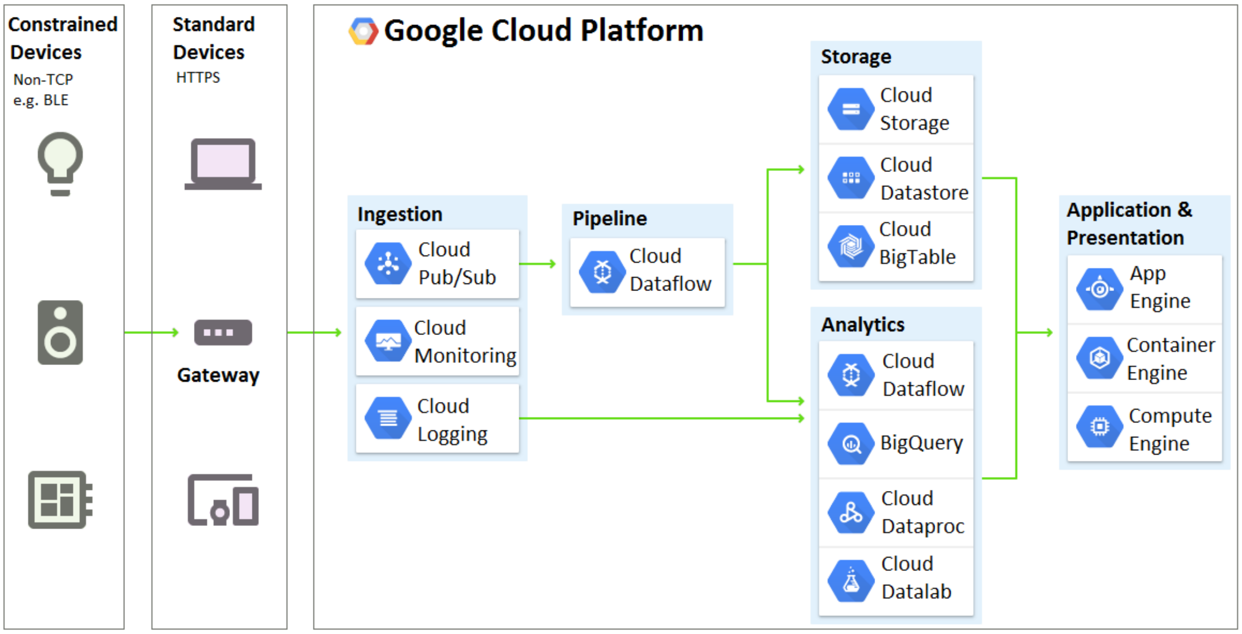


Figure 19: An overview of Google cloud platform for IoT, adapted from (Google, n.d.-c)

- **Ingestion:** Information from devices can be imported into the cloud platform services using the process called Ingestion. There are different Ingestion services support in the Google cloud platform depending on the type of information (telemetry or operational information) coming from the device:
  - **Cloud Pub/Sub:** Cloud Pub/Sub is a real-time messaging service and allows to communicate between the independent applications. To publish and subscribe to data from multiple sources, cloud Pub/Sub can be used. It is highly scalable (10000 messages per second) and guarantees the delivery of the message.
  - **Cloud monitoring:** The collected data from cloud platform generates insights via dashboard, charts, and alerts.
  - **Cloud logging:** It is a logging functionality that collects and stores logs to cloud platform to view, search, filter, and export.

- **Pipeline:** Pipelining is used to transform, aggregate, and compute the data received from Ingestion module. The cloud dataflow in this stage has a programming model that can be used to manage high volume-data.

- **Storage and analytics:** The data can be stored in the database and analysed and visualize using analytical tools.

- **Application and presentation:** Application and presentation can be supported by App engine, container engine, and compute engine (see Figure 19).

4.1.4 Microsoft: Azure IoT Suite

This section gives a brief introduction to Microsoft's IoT approach.

A Brief Company Profile

Microsoft (https://www.microsoft.com)	
A brief introduction	Microsoft (Nasdaq 'MSFT'), founded in 1975, provides software, services, devices and related solutions. Microsoft has released its cloud computing platform, Microsoft Azure, on 1st of February, 2010 (Microsoft, 2010). Microsoft Azure provides different cloud services including analytics, computing, database support, mobile, networking, storage, and the web. Azure IoT Suite together with Azure services (Azure IoT Hub, Azure event hubs, Azure stream analytics, Azure machine learning, and Azure storage) makes it possible to support a broad range of capabilities: data collection from devices, data analysis, data storage, real-time and historical data visualization, and integration with back-office systems. A core component of Azure IoT Suite is IoT Hub that provides the messaging capabilities between devices and cloud system.
IoT platform	Azure IoT suite.
Industrial partners in IoT	IoT partners (Microsoft, 2016-a): **Device partners:** AAEON, Acme Systems, Adafruit, ADLINK Technology, Advantech, Alleantia S.r.l., Arvor Technology Corp, Arduino, Arrow Electronics, Avalue Technology, Axiomtek, Beagleboard, Bluebird, Clientron Corp, ComponentSoft, Contec Co, DFI Inc., DUX Inc., Embedded Systems, Fitivision Technology, Freescale, Fujitsu Limited, Hitachi Industry and Control Solutions, IEI Integration Corp, Intel, Inventec Corp., Keith & Koep, Lanner Technology, Libelium, MechaTrax, Mert Yazilim Bilgisayar (TrexDCAS), MinnowBoard, MiTAC Computing Techonology Corp, NEXCOM International, Panasonic, Partner Tech, PFU Limited, Raspberry Pi, Resin.io, Ricoh Industrial Solutions Inc., Samsung, Seeed, Seraku Co. LTD., SOTEC, Sparkfun, STMicroelectronics, Texas Instruments, Toradex, Toshiba Platform Solution Corporation, Universal Global Scientific Industrial, ZEROTECH Intelligence Technology Co. Ltd., **Gateway partners:** Acme Systems, Alleantia S.r.l., Amplified Engineering, Atmark Techno Inc., Axiomtek, Beckhoff Automation, Contec Co, Dell, DUX Inc, Ecomott, E-con Systems, Fitivision Technology, HP, IEI Integration Corp, Intel, Lanner Technology, Libelium, Pacific Control

Microsoft (https://www.microsoft.com)	
Industrial partners in IoT	Systems, Plat'Home Co, SADE Group, Smarthesia, Universal Global Scientific Industrial, Vantron Technology, Yaskawa Informations Systems.
Awards or nominations	• Lifehacker awards: Microsoft Azure won the best cloud computing provider award (editor's choice) 2014 (Kidman, 2014).

Table 5: Microsoft – A brief profile

Azure IoT Suite

Microsoft Azure IoT solution architecture is shown in Figure 20. The cloud gateway receives the data from IoT devices (IoT client) directly or through a gateway (IoT client) depending on the network capabilities of IoT device. This device data can be stored or pass for further processing, analytics, and management. The processed data can later be delivered to other business and presentation applications.

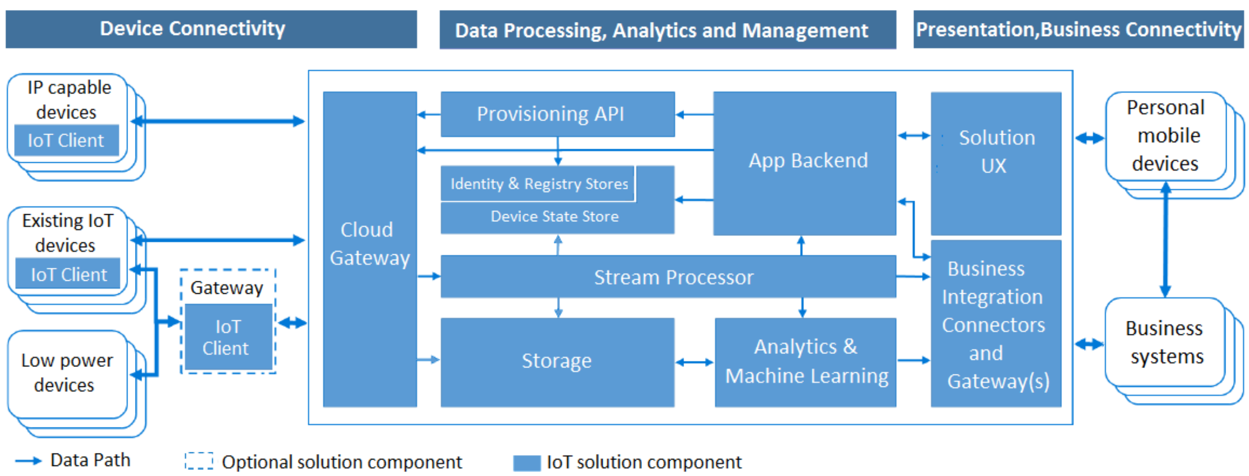


Figure 20: Azure solution architecture, adapted from (Betts, 2016)

Azure IoT hub, Figure 21, provides reliable and secure bidirectional communication between IoT solution backend and IoT device. IP-enabled IoT devices can communicate with the IoT Hub over HTTP, MQTT, or AMQP protocol. Other devices can communicate to IoT Hub through an IoT protocol gateway.

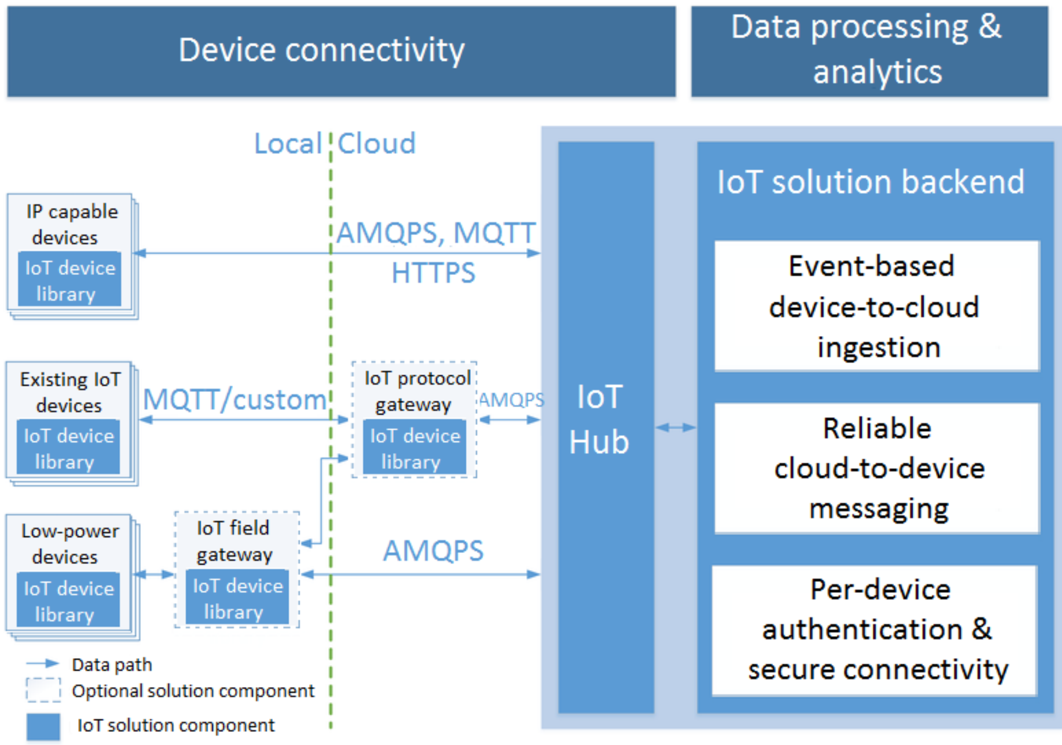


Figure 21: Azure IoT hub, adapted from (Betts, 2016)

IoT Solution backend (see Figure 21) has an event processor engine for processing millions of events per second received from the device. To guarantee reliable processing, IoT Hub retains the event data for up to seven days.

4.1.5 Amazon: AWS IoT

This section gives a brief introduction to Amazon's IoT approach.

A Brief Company Profile

Amazon (https://www.amazon.com/)	
A brief introduction	Amazon.com, Inc., incorporated on 28th of May, 1996, has three segments: North America, International, and Amazon Web Services (AWS) (Reuters, n.d.-b). Since 2006, AWS (Amazon, n.d.-b) has been offering IT infrastructure services as cloud computing which includes database storage, networking, content delivery etc. AWS's IoT is a managed cloud platform that enables us to connect devices and interact with cloud applications and other services. AWS IoT has a support for billions of devices and trillions of messages. With AWS IoT you can also use other AWS services like AWS Lambda (a compute service that can run your uploaded code using AWS infrastructure), Amazon Kinesis (used to collect and process a large stream of data in real time), Amazon S3 (a simple storage service), Amazon Machine Learning etc.
IoT platform	Amazon AWT IoT.
Industrial partners in IoT	The AWS Partner Network (APN) program (Amazon, n.d.-c) helps their global partners in building a successful AWS-based business solution. The AWS IoT hardware partners provide products supported by AWS IoT services. Some of the hardware partners that provide featured AWS IoT starter kits are: Avnet Cloud Solutions, Intel, Broadcom, Microchip Technology, Marvell, Arrow Technology, Micrium, Mistral Solution Pvt. Ltd., Mediatek, Renesas, Texas Instruments and Seeed Studio.
Awards or nominations	• Nominated for people's choice award 2015/16 for IoT cloud platform (Postscapes, 2016).

Table 6: Amazon – A brief profile

AWS IoT

Amazon AWS is primarily known as an Infrastructure as a Service (IaaS). But, it has many services similar to that in any PaaS platform. The AWS IoT is one of the services of AWS that enables the communication between the device and many AWS services. The core components of AWS IoT are shown in Figure 22.

- AWS IoT SDKs are freely available as an open-source that helps in connecting the hardware devices with AWS IoT.

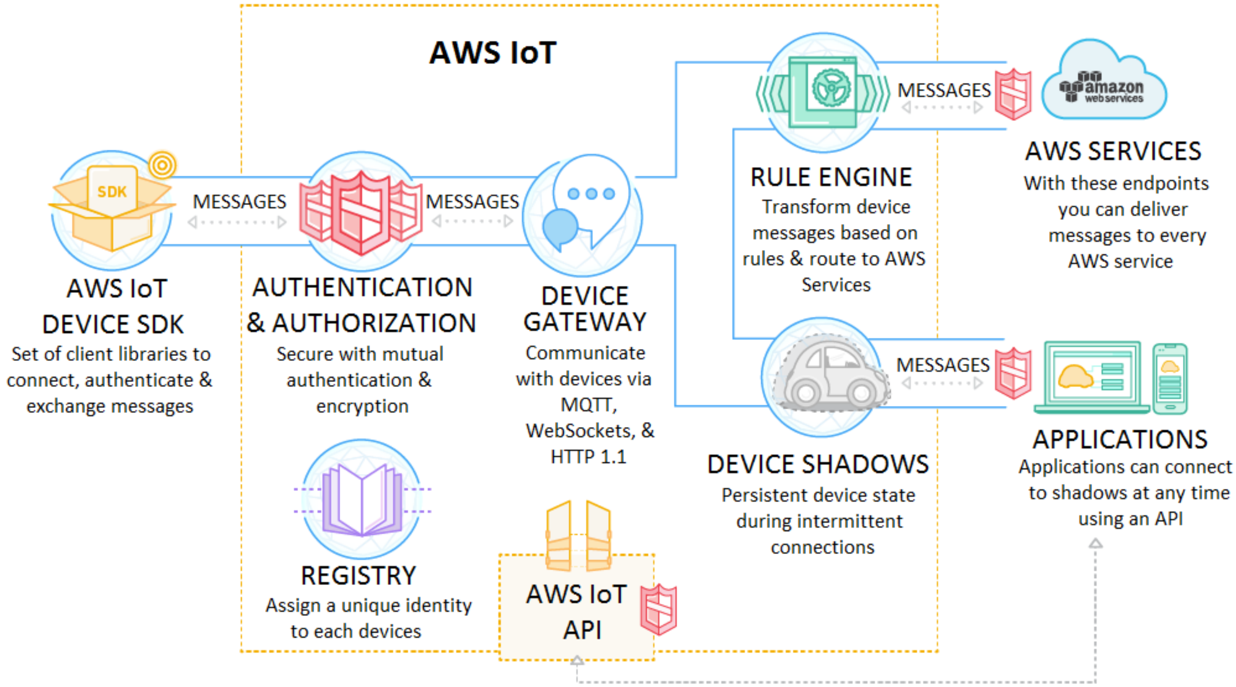


Figure 22: AWS IoT features, adapted from (Amazon, n.d.-d)

- AWS IoT rule engine uses an SQL-like syntax to select data from MQTT client, to perform actions on the data, to create rules, and to create policies. A list of AWS IoT API reference is available (Amazon, 2016) that can be used to build the application. Rule engine makes it possible to build IoT applications that can gather, analyse, process, and act on the data received from the device.

- AWS IoT rule engine can also route the message to AWS endpoints including AWS lambda, Amazon kinesis, Amazon s3, Amazon machine learning, Amazon dynamoDB, Amazon cloudWatch, Amazon elasticSearch service.

The device shadow needs to be updated once in a year, otherwise, it will expire.

4.2 Comparison of IoT Development Platforms

This section compares various IoT development platforms discussed above on the basis of general, business and technical criteria. The following color scheme has been used in the comparison table.

Yes, feature is provided by the IoT platform developer.

Yes, feature is available through the partners or third party.

No, feature is not available.

No information is available.

4.2.1 Comparison Criteria and their Description

The comparison criteria have been classified into two parts:

1. General and business aspects
2. Technical aspects

The descriptions of the different comparison criteria used in this report are presented below. These criteria are used to compare the different IoT development platforms which were presented in the previous sections.

Table 7 gives a description of general and business criteria used to compare IoT platforms.

General and business aspects	
Comparison criteria	**Description**
Important events in IoT development	Important dates in the Internet of Things development process. For example, release date of the Internet of Things platform, other important releases/announcements related to IoT.
Regions (data centres)	The regions are defined geographical locations where the IoT applications are being stored and deployed. Irrespective of the current geographical location, the IoT services can be used if access to these defined regions is provided.
Free trial	Whether free trial is available or not. If available, what are the free usage offers?
Pricing method(s)	The pricing methods used by cloud-based IoT development platforms. For example, pay per device, pay per message, etc.

General and business aspects	
Comparison criteria	**Description**
Support plan(s)	Is there any support plan available for the IoT platform users? If yes, what are the features and offers?
Application market place	Is there any Application Market Place available? A Marketplace provides access to download applications, software, etc. to help you in building your application. The service providers and/or partners of IoT platform providers can publish their services at the Market Place.
Availability of documents	Are there any documentations, audio/video tutorials, online courses etc. available?
Community support	Is there any online community support available to discuss the issues, report problems etc.?

Table 7: Criteria - General and business

Table 8 gives a description of technical criteria used for the comparison of IoT platforms.

Technical aspects	
Comparison criteria	**Description**
Architecture	IoT develop platform architectural approach (i.e. cloud based).
PaaS platform	Name of the Platform as a Service (PaaS) platform.
Proprietary or open source	Is the solution is based on open source, or it is proprietary?
Platform access option	How do you access the IoT service platform? For example, GUI (Graphical User Interface), CLI (Command Line Interface) etc.
Protocol	Name of the supported protocols.
Data format supported	Data format used to communicate between the devices and IoT development platform.
Storage	Is there any feature to store the data? If yes, name the feature(s).

Technical aspects	
Comparison criteria	**Description**
SDKs/Languages	Is there any SDK (Software Development Toolkit) available? If yes, then in which language(s)? What are the other languages supported by the IoT development platform to build an application?
Transport security	Which type of transport layer security is supported?
Authentication	Types of authentication supported (e.g. application key, etc.).
Support to create rules	Is it possible to set the rules to collect, manage, process, and act on the data coming from the client applications or devices?
Visualization of data	Does the IoT development platform provide visualization of the data in graphical form? For example, historical or real-time graphs etc.
Development tool support	Is there a possibility to use some development tools (e.g. eclipse) to implement your application locally and upload it into the IoT development platform?
Debugging or logging availability	Does the IoT development platform provide debugging tools or generate run-time traces (logs) to track the status of application program?
Deployment options	What are the deployment options? For example, cloud, on premise, hybrid, or on-device deployments.
Social network linkage	Is it possible to link any social network (e.g. Twitter, Facebook, etc.) to the application, or service running using the IoT development platform?
Source code management	Is there any option to manage the source code using some in-built or online external repositories?

Table 8: Criteria - Technical

4.2.2 Comparison of Platforms

In this section, the criteria discussed in Table 7 and Table 8 to compare different IoT development platforms are applied. **Error! Reference source not found.** and Table 10 show the comparison on the basis of general and business aspects and technical aspects, respectively.

General and business aspects					
Criteria	**Watson IoT for Bluemix**	**ThingWorx**	**Google Cloud Computing IoT**	**Azure IoT Suite**	**Amazon Web Services (AWS) IoT**
Important events in IoT development	• On 20th of June, 2014, IBM announced Open Cloud Development platform (IBM, 2014). • On 15th of October, 2014, IBM's Internet of Things for Bluemix was launched.	• In 2011, ThingWorx platform was released. • In 2014, PTC acquired ThingWorx. • In 2014, PTC acquired Axeda.	• In August, 2015, Google's cloud Pub/Sub real-time messaging service was launched (Eric Schmidt, 2015). • In May, 2015, Google announced Brillo, an operating system for the Internet of Things in Google I/O conference.	• On March 16th, 2015, Azure IoT Suite was announced during Microsoft convergence in Atlanta (Numoto, 2015).	• In October,2015, AWS IoT feature was announced in AWS re:Invent conference in Las Vegas (Amazon, 2015).
Regions (data centres)	3 Regions • US South • United Kingdom • Sydney	ThingWorx IoT servers are available out of all Amazon AWS regions. Also, it uses some	3 Regions (Google, n.d.-d) • Central US • Eastern US • Western Europe • Eastern Asia	9 Regions (Microsoft, 2016-b) (Singh, 2016) • East Asia • East US • North Europe	6 Regions (Amazon, n.d.-d) • US East (N. Virginia) • US West (Oregon) • EU (Ireland)

General and business aspects					
Criteria	**Watson IoT for Bluemix**	**ThingWorx**	**Google Cloud Computing IoT**	**Azure IoT Suite**	**Amazon Web Services (AWS) IoT**
Regions (data centres)		private data centres in North America and EU regions.		• Southeast Asia • West Europe • West US • Australia • Japan • Germany (preview)	• EU (Frankfurt) • Asia Pacific (Tokyo) • Asia Pacific (Singapore)
Free trial	● 30 Days free trial **Features:** 20 free active devices, 100 MB of free data traffic, 1 GB of free data storage, 20 included devices.	● 30 Days free trial No other information is available.	● 60 Days free trial for Google cloud platform and all the related services. **Features:** $300 credit to spend on Google cloud platform over 60 days of free trial.	● IoT Hub free trial is for unlimited duration. **Features:** Transmits up to a total of 8,000 messages a day, registers up to 500 devices.	● 12 Months free trial. AWS calls it free tier. **Features:** 250,000 free messages (published or delivered) per month, for 12 months.
Pricing method	The IBM Watson IoT platform charges on three metrics: • The average	The subscription option is used to purchase any PTC product including ThingWorx.	Google cloud service's pricing model is based on per minute billing and automatic	IoT Hub service is charged on the basis of IoT Hub units consumed. The consumption	The AWS IoT charges on usage basis and there are no minimum fees for this service. The

General and business aspects					
Criteria	**Watson IoT for Bluemix**	**ThingWorx**	**Google Cloud Computing IoT**	**Azure IoT Suite**	**Amazon Web Services (AWS) IoT**
Pricing method	number of devices connected in a month. • The amount of data exchanged between devices and IBM Watson IoT platform and associated applications. • The amount of data stored in historical database.	Subscription enables you to pay for what you need, and when you need it.	discount as the usage increases.	of IoT Hub messages is measured daily and billed on monthly basis.	pricing is based on publishing cost (the number of messages published to AWS IoT), and delivery Cost (the number of messages delivered to devices or applications by AWS IoT). There is no charge for delivering to the following AWS services: Amazon S3, Amazon DynamoDB, AWS Lambda, Amazon Kinesis, Amazon SNS, and Amazon SQS.
Support plan(s)	● IBM IoT foundation	● • Gold (availability	● • Bronze (free of	● IoT Hub is available	● AWS support plan is

General and business aspects					
Criteria	**Watson IoT for Bluemix**	**ThingWorx**	**Google Cloud Computing IoT**	**Azure IoT Suite**	**Amazon Web Services (AWS) IoT**
Support plan(s)	offers bronze, silver and gold plans. In addition to free plan offers, the package details are following: • Bronze (€ 19.30 EUR): 100 included devices, € 0.96 per additional 100 devices, € 0.96 per additional 100 MB of data traffic, € 0.96 per additional 1 GB of data storage. • Silver (€ 116 EUR): 3000 included devices, € 3.86 per additional 100 devices, € 0.96 per additional 100 MB of data traffic, € 0.96 per additional 1 GB of data	24x5) without onsite support • Gold Plus (availability 24x7) without onsite support • Platinum (availability 24x7) with 6 days' onsite support	charge with the very basic support). • Silver ($150 per month): It includes Architecture support for best practice etc. • Gold (starts at $400 per month): User case specific architectural support, phone support, 24x7 support for critical impact issues, etc. • Platinum (pricing not available): User case specific architectural support, phone support, 24x7 support for critical impact issues, 24x7 support for high	in three editions: • Free (limit: 8000 messages/day, message meter size: 0.5 KB) • S1 (subscription: $50 per month, limit: 400,000 messages/day, message meter size: 4 KB) • S2 (subscription: $500 per month, limit: 6,000,000 messages/day, message meter size: 4 KB)	divided into four categories: • Basic (free) • Developer ($49/month) • Business (starting $100/month) • Enterprise (starting $15000/month) Documentations, whitepapers, best practice guides can be accessed with all the available plans.

General and business aspects					
Criteria	**Watson IoT for Bluemix**	**ThingWorx**	**Google Cloud Computing IoT**	**Azure IoT Suite**	**Amazon Web Services (AWS) IoT**
Support plan(s)	storage. • Gold (€ 434 EUR): 15000 included devices, € 2.89 per additional 100 devices, € 0.96 per additional 100 MB of data traffic, € 0.96 per additional 1 GB of data storage		impact issues, etc.		
Application market place	● • IBM marketplace	● • ThingWorx marketplace • PTC marketplace	● • Google Apps marketplace	● • Azure marketplace	● • Amazon market place.
Availability of documents	● Tutorials, videos, online courses, IoT recipes, etc. are available.	● Documentation, video tutorials, online courses, etc. are available.	● Tutorial are available for most of the cloud services. For information on	● Tutorials, whitepapers, videos etc. are available.	● Developers guide, API references, webinars, slides, videos etc. are available.

General and business aspects					
Criteria	**Watson IoT for Bluemix**	**ThingWorx**	**Google Cloud Computing IoT**	**Azure IoT Suite**	**Amazon Web Services (AWS) IoT**
Availability of documents			Brillo and Weave, you need to request for an Invite.		
Community support	• IBM DeveloperWorks (from IBM) • StackOverflow (Third party)	• ThingWorx Developer Community (from ThingWorx)	• Google+ (Official Google Cloud Platform Community). • StackExchange (i.e. StatckOverflow and ServerFault). (Google, n.d.-e) (Third party)	• MSDN forum (from Microsoft) • StackOverflow (Third party)	• Amazon Web services Discussion Forum (Amazon, n.d.-e) (from Amazon)

Table 9: Comparison - General and business aspects

The following table compares the technical features of different IoT platforms.

Technical aspects					
Criteria	**Watson IoT for Bluemix**	**ThingWorx**	**Google Cloud Computing IoT**	**Azure IoT Suite**	**Amazon Web Services (AWS) IoT**
Architecture	Cloud based	Cloud based	Cloud based	Cloud based	Cloud based
PaaS platform(s)	IBM Bluemix	ThingWorx	Google Cloud Platform	Microsoft Azure	AWS
Proprietary or open source	Proprietary Bluemix is based on Cloud Foundry open source.	Proprietary	Proprietary	Proprietary	Proprietary
Platform access options	• Console (GUI) • CF (Cloud Foundry) CLI	• ThingWorx composer (GUI)	• Google cloud platform console (GUI) • Cloud shell CLI • Weave console (GUI)	• Azure portal (GUI) • IoT Hub explorer (CLI)	• AWS management console (GUI) • CLI
Protocol support	• HTTP • MQTT • MQTT over WebSockets	• HTTP Extension protocol adapters (ThingWorx, n.d.-c) • MQTT • RabbitMQ	• HTTPS • Weave cross platform protocol (Note: It is different than Nest's Weave protocol)	• AMQP • AMQP over WebSockets • MQTT • HTTP/1	• HTTP • MQTT • MQTT over WebSockets

Technical aspects					
Criteria	**Watson IoT for Bluemix**	**ThingWorx**	**Google Cloud Computing IoT**	**Azure IoT Suite**	**Amazon Web Services (AWS) IoT**
Data format supported	JSON	Not specified	JSON	JSON	JSON

Storage	• Cloudant NoSQL DB
	• dashDB
	• IBM DB2 on cloud
	• MongoDB by Compose
	• SQL Database
		Following storage options are available (ThingWorx, n.d.-b):
		
		• Data Tables
		• Streams
		• Value Streams
		
		• external Databases
		  • DataStax Enterprise
		  • Neo4j
		  • PostgreSQL
			• Cloud Bigtable
			• Cloud Storage
			• Cloud DataStore
				To store device data, we have following options (Basak, 2015):
				• SQL database
				• Blob storage
				• Table storage
				• DocumentDB
					AWS IoT rule engine can directly communicate with the Amazon database services:
					
					• Amazon S3 (A Simple Storage Service)
					• Amazon DynamoDB (NoSQL database service)
					• Amazon Kinesis (collect and process real-time stream data)

SDKs/Languages	Language support (IBM, 2016-a): Application development: • HTTP • Python • Node.js	• Java • .NET • C SDK • iOS SDK	• Google Cloud SDK (Google, n.d.-f). Pub/Sub client libraries are available in Go, Node.js, Python and Ruby.	Device SDKs available in following languages: • C • C# • .NET • Java	Device SDKs available in following languages: • Embedded C • JavaScript/ Node.js • Arduino Yún platform

Technical aspects					
Criteria	**Watson IoT for Bluemix**	**ThingWorx**	**Google Cloud Computing IoT**	**Azure IoT Suite**	**Amazon Web Services (AWS) IoT**
SDKs/Languages	• Java • C# Device development: • HTTP • Python • Node.js • Java • C# • Embedded C • mBed C++		• Brillo is buildable from Android Open Source Project (AOSP) (Trebilcox-Ruiz, 2016). Language for development: C, C++.	• Node.js • Python 2.7 Application service SDK: • Node.js • Java. IoT gateway SDK is also available at GitHub.	• iOS
Transport security	TLS	TLS	TLS	TLS	TLS
Authentication	• API Key and authentication token. • User defined token	• Application Key	Device can be provided any of the following for a secure communication: • Token • Service account key	• Token based authentication. IoT Hub supports authentication using X.509 certification for devices over AMQP, AMQP over web socket, and HTTP.	Three types of identity principals are used for authentication: • X.509 certificates • IAM users, groups, and roles • Amazon Cognito identities For HTTP protocol, use IAM, or Amazon Cognito.

Technical aspects					
Criteria	**Watson IoT for Bluemix**	**ThingWorx**	**Google Cloud Computing IoT**	**Azure IoT Suite**	**Amazon Web Services (AWS) IoT**
Authentication					For MQTT protocol, use X.509 certificate.
Support to create rules	Node.js application is used to create rules of the data received from the applications or services.	Expression widget.	Google Cloud Function allows to write custom logics.	Azure Stream Analytics (ASA) is used to develop the real time streaming solution for IoT. We can define the alert rules based on the input events captured by ASA.	AWS IoT has 'Rule Engine' that makes it possible to build an IoT application that gathers, analyses, process, and acts on the data received from the smart devices.
Visualization of data	IBM Watson IoT real-time insight can be used to visualize IoT data etc.	ThingWorx composer allows to combine data service using a set of visualization components, called Widgets.	Cloud Datalab can be used for visualization.	Azure Web Apps and Microsoft Power BI provides Data visualization capabilities.	AWS IoT Rule Engine can route the data to Amazon Elasticsearch (Amazon) which can later be visualize using the tool like Kibana (Elastic).

Technical aspects					
Criteria	**Watson IoT for Bluemix**	**ThingWorx**	**Google Cloud Computing IoT**	**Azure IoT Suite**	**Amazon Web Services (AWS) IoT**
Development tool support	• Eclipse • Visual Studio	• Eclipse	• Cloud SDK • Google plugin for Eclipse • Cloud tools for Android Studio • Cloud tools for IntelliJ • Brillo Development Kit (BDK)	• Microsoft Visual Studio	• Eclipse Paho for JavaScript.
Debugging or logging availability	Logging is an inbuilt functionality in IBM Bluemix that can be accessed through dashboard. Also, Node.js debugging is a very useful feature.	• Application logs • Communication • Composer • Configuration	• Google cloud logging • Google cloud monitoring • Google cloud audit logs • Google Stackdrive error reporting	• Preconfigured solutions can be used for logging.	CloudTrail is a service that is integrated with AWS IoT and captures AWS IoT API calls and delivers the logs to Amazon S3 (Amazon, n.d.-f).

Technical aspects					
Criteria	**Watson IoT for Bluemix**	**ThingWorx**	**Google Cloud Computing IoT**	**Azure IoT Suite**	**Amazon Web Services (AWS) IoT**
Deployment options	• Public • Dedicated (on public cloud) • On-premises (on your own cloud)	• On-premises • Embedded • Hybrid		• Public • Private • Hybrid But, main guidance focus is on Public cloud implementation.	
Social network linkage	• Twitter • Email	• Twitter • Amazon Payment • Facebook • Google+			
Source code management	• Built-in support for Jazz source control • Hosted Git repository • External Git Hub	• FileRepository	• Google cloud source repository (Google, n.d.-g).	• Data Lake • GitHub	• AWS CodeDeploy supports GitHub.

Table 10: Comparison - Technical aspects

4.2.3 Summary/Comments

Google, Microsoft, and Amazon have recently (in the year 2015) launched their IoT solution, IBM's Watson IoT is on the market since 2014. The expectations from these IoT solutions is high because they are built upon their already existing cloud platforms and services. ThingWorx, launched in 2011, provides a complete IoT solution. Table 11 shows a short summary of comparison done above. As can be seen, all the platforms provide the basic IoT functionalities.

ThingWorx, a model based platform, allows you to create new widgets or use already available widgets to develop the IoT applications. ThingWorx servers are hosted on third party data centres (i.e. Amazon AWS) and some other private data centres.

Google cloud platform for IoT allows the data from devices to be received using its Ingestion module (using Pub/Sub API's). Google has recently launched Brillo, an operating system for IoT devices. Together with Brillo, Weave technology (cross platform) and Google cloud platform, Google is going to offer a very good IoT solution. But, to use this solution one needs to have Brillo compatible devices. Also, there is no support for commonly used MQTT or AMQP protocol in the Google's IoT solution.

Microsoft and Amazon offer a free trial for a much longer duration (unlimited and 12 months respectively) than the other platforms discussed in this report. To use the free trial of all these platforms (except for IBM Bluemix) one needs to provide our banking information.

Criteria	Watson IoT for Bluemix	ThingWorx	Google Cloud Computing IoT	Azure IoT Suite	Amazon Web Services (AWS) IoT
Flexible pricing	+	+	+	+	+
Data centre	+	(+)	+	+	+
Service support	+	+	+	+	+
GUI interface	+	+	+	+	+
MQTT support	+	+	-	+	+
Device SDK	+	+	+	+	+
Storage	+	+	+	+	+
Secure communication	+	+	+	+	+
Rules creation	+	+	+	+	+
Data visualization	+	+	+	+	(+)

Criteria	Watson IoT for Bluemix	ThingWorx	Google Cloud Computing IoT	Azure IoT Suite	Amazon Web Services (AWS) IoT
Local development	+	+	+	+	+
Source code management	+	+	+	+	+
Integrated runtime environment	+	-	-	-	-
Legends: Feature is available: **+** Feature is not available: **-** Feature is available from third party: **(+)**					

Table 11: Comparison - Summary

IBM's Watson IoT for Bluemix appears to be the best choice among the IoT platforms discussed in this research report. IBM's Watson IoT is integrated into the Bluemix which is an implementation of IBM's Open Cloud Architecture. IBM's Open Cloud is based on the Cloud Foundry which is an open source PaaS and enables you to quickly create and deploy applications on the cloud. IBM Watson IoT provides multiple language support to implement the IoT solution. The integration of runtimes (e.g. Node.js) gives it an edge over most of the other IoT development platforms. IBM has a large number of partners to support its IoT solution. There are many features of Bluemix that make it a good choice for developing IoT-based applications, for example, one can edit the source code within the Bluemix environment without installing any tool in the local system, code can be stored using a Built-in Jazz source control, Bluemix's hosted Git repository, or external GitHub. Plugin for eclipse is available that helps in implementing, compiling, debugging the code locally and later uploading it back to the Bluemix using cloud foundry CLI interface. The sensor data can be posted to a Twitter account or send it to the inbox via email by just providing the login credentials to the built-in Node.js node. The Watson service for Bluemix also provides support to build cognitive applications. The cognitive capabilities of Watson include natural language processing (NLP), machine learning, text analytics and video/image/audio analytics. IBM Bluemix's capabilities are discussed in more detail in next section.

4.3 IBM Bluemix

IBM Bluemix is an open standard cloud-based Platform as a Service (PaaS) that runs over Softlayer which is IBM's worldwide Infrastructure as a Service (IaaS) (IBM, 2016-b). IBM Bluemix offers three deployment methods: Public, Dedicated (on the public cloud), and On-premise (on your own data centre). Bluemix's DevOps service, a software development method, emphasizes communication, collaboration, integration and automation. Using DevOps, one can online plan, develop, and test an application. It uses the pipelining core concept (Figure 25) of Build-Test-Deploy. The project can be made public or be saved with private access. There is also a possibility to invite users to the project. Bluemix provides built-in software management, and online editor to develop an application. The structure of IBM Bluemix is shown in Figure 23.

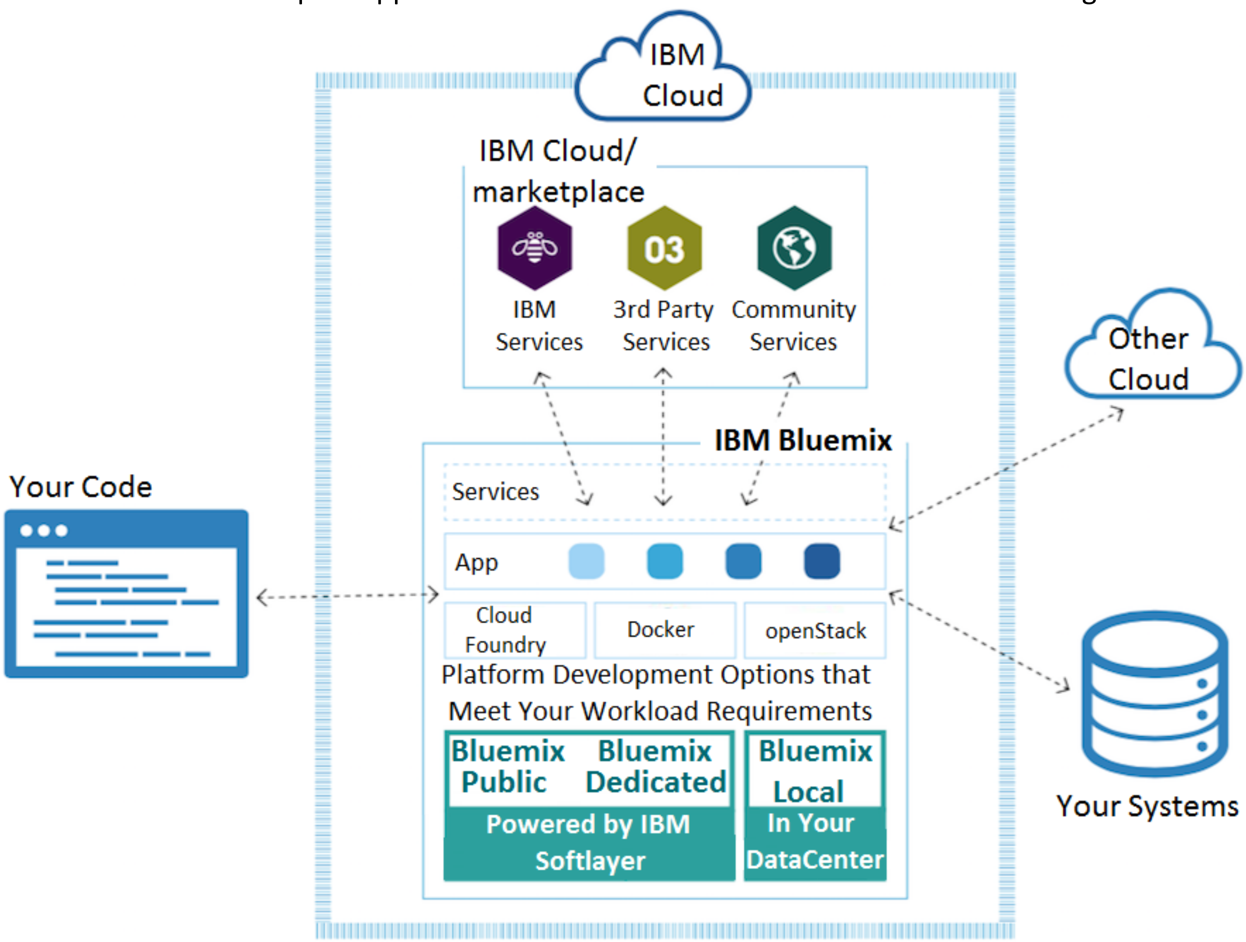


Figure 23: IBM Bluemix structure, adapted from (IBM, 2016-b)

Bluemix, based on cloud foundry, has multiple programming language support including Java, Node.js, Go, PHP, Python, Ruby Sinatra, Ruby on Rails, Scala (extended through build pack). Bluemix also offers different compute models to run the code: Cloud Foundry, Docker, OpenStack. Docker runs on the host machine and uses the resources of that machine. IBM Bluemix also provides a number of database and database services: No SQL database service (e.g. Cloudant, MongoLab), SQL database service (e.g. DB2, ClearDB, Elephant SQL), Key Value Pair database service (e.g. Redis Cloud, IBM Data Cache), Time-series database.

In Bluemix's PaaS approach, Bluemix's client manages the applications and data, while vendor (Bluemix) manages runtime, middleware, o/s, virtualization, server, storage and networking. Figure 24 shows the high-level public PaaS architecture of Bluemix. Developers can use Bluemix's browser-based interface to interact with the Bluemix services. Also, the Cloud Foundry command line (CL) interface can be used to deploy the web applications. As shown in Figure 24, Bluemix's client (i.e. mobile application client, web application clients, application developers) can use REST or HTTP APIs to route requests to Bluemix application instance or the composite service.

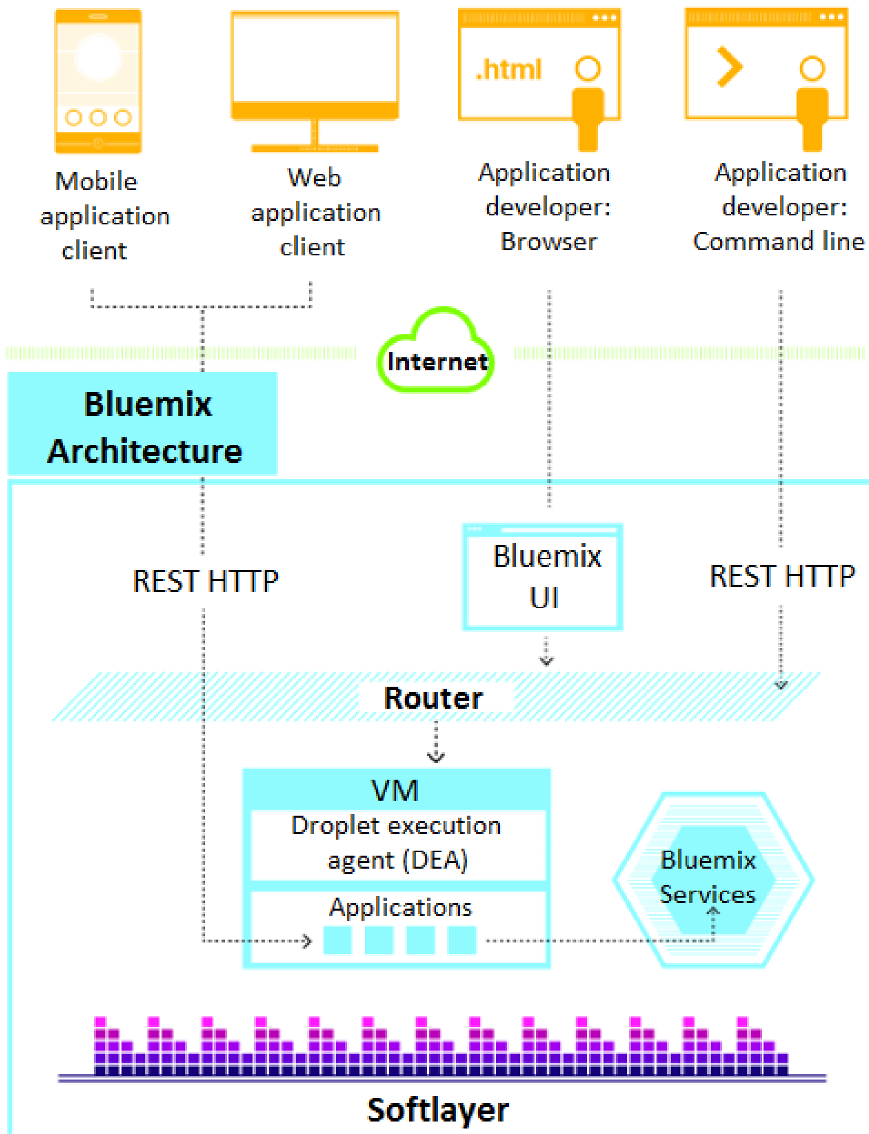


Figure 24: IBM Bluemix architecture, adapted from (IBM, 2016-b)

The IBM Bluemix uses the following terminologies:

- **Space:** A space in Bluemix is a logical grouping of all the resources that you consume in Bluemix.

- **Boilerplate:** Boilerplate provides the quick links of applications and the associated run-time environment. It also predefines the services for a particular domain.
- **Service:** Bluemix provides a large number of services that can be chosen while developing a web or mobile application. The Internet of Things platform service can be used to manage and control IoT devices. Other services like Watson (to build cognitive applications), database service etc. can be combined with the IoT platform services to build advanced applications.
- **Application:** IBM Bluemix provides a run-time environment to build the web-based or mobile-based application. Bluemix supports multiple languages, as discussed before, that can be used to build interactive applications.

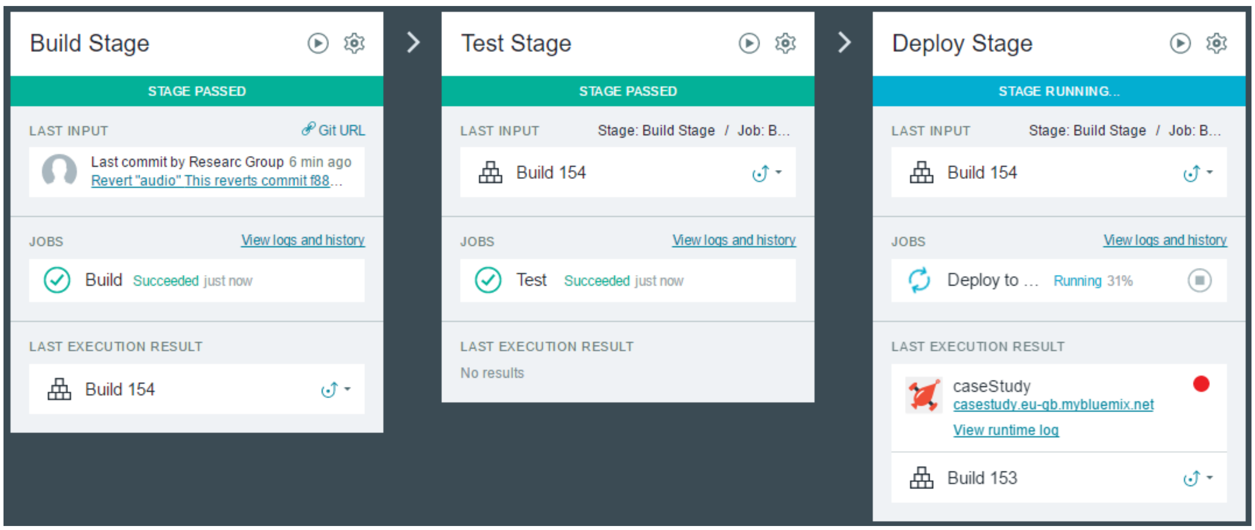


Figure 25: Pipeline core concept in IBM Bluemix to build, test and deploy

IBM Bluemix support is available through online communities such as Stack Overflow and IBM developer support site.

4.4 Watson IoT for Bluemix

IBM Watson for IoT is available through the IBM Bluemix platform. IBM Bluemix enables you to create web-based and mobile based applications in different languages including PHP, java, node.js. Bluemix's IoT service receives the data from the smart devices and can send it to the applications bind to it. IBM Watson IoT for Bluemix offerings include analytics, risk management, connect and information management (see Figure 26). One can connect the IoT enabled devices using MQTT protocol to the Watson IoT platform which also supports device management and visualization of real-time data. To gain the insight from the real-time data coming from the IoT devices, Watson IoT provides predictive, cognitive and contextual analytics. The IBM Watson IoT cloud risk management feature ensures to provide right and up-to-date information. IBM Watson IoT information management provides structured and unstructured data management, information storage, and information archiving.

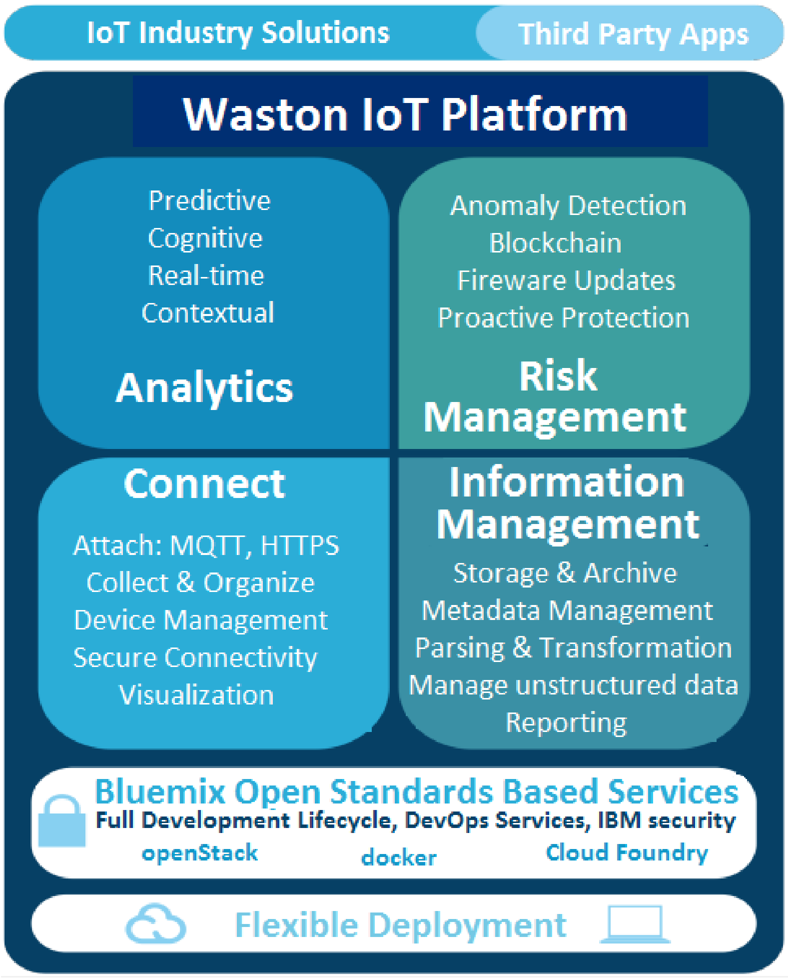


Figure 26: IBM Watson IoT platform offerings, adapted from (Caldwell, 2015)

IBM Watson IoT fulfils the fundamental requirements of IoT as mentioned above such as connecting "things", collecting real-time data from "things", managing "things", data storage, and data analytics. IBM with its partners (refer section 4.1.1) is working in the direction to define the new innovations in IoT field. IBM IoT platform is effective in providing the IoT solutions in different sectors such as healthcare, manufacturing industry, transportation and retail sector (Caldwell, 2015). Some of the use cases developed using the IBM's IoT platform are discussed in section 4.4.3.

The next section explains the connectivity between the different modules and steps to register the device and send, process data using IBM Bluemix IoT services.

4.4.1 Communication between IoT Device, IBM IoT Platform, and Application

An IoT device can be connected to the IBM's cloud-based IoT platform through the MQTT protocol. IBM Watson IoT for Bluemix allows its users to create web-based or mobile-based applications. The communication between the IoT platform and Application can be possible through the HTTP protocol over web-socket.

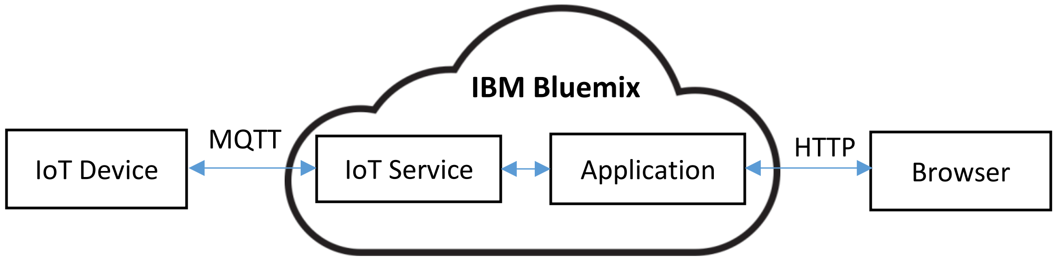


Figure 27: Communication b/w IoT device, IBM IoT cloud-based platform, and application, own illustration

The following are the steps needed to enable communication between the IoT device and Bluemix IoT platform, and between Bluemix IoT platform and the IoT Application.

1. Log in to the IBM Bluemix platform's GUI Interface (e.g. platform's console).
2. Register the device(s) on the IoT Platform. This step will generate the unique identification for the device at the IoT platform. Device authentication options (i.e. access key, user-defined token etc.) can be selected in this step.
3. Connect device(s) with the IoT platform using MQTT protocol. Use the access options decided in step 2.
4. After successful connection, the device can send the data to the connected IoT platform.
5. The IoT platform can now receive the data and takes action if the data is in JSON format.
6. Data generated from the devices can be visualized and analysed, or can be sent to the associated application for further action.
7. The authorised application can send back the command to IoT service running at the IoT platform to control the device. The IoT device will send it to the concerned device to perform an action on the command.
8. IoT platform service can have predefined rules to take action on the device data, for example, if a device is sending the temperature data, the predefined rule (i.e. trigger on exceeding a threshold limit) can send back the control command to the device.

A concrete use case, shown in the next chapter, is developed by using the IBM Watson IoT for Bluemix, Intel Edison kit and Grove indoor environment kit. The use case uses the various features available in Bluemix's IoT solution. The digital artifacts like PHP application and Node.js application are created using Bluemix.

4.4.2 Visualization of Data in IBM Watson IoT for Bluemix

The data collected from the IoT devices can be visualized using the in-built visualization application in Bluemix. The Internet of Things service of Bluemix allows you to create real time charts to visualize the sensor data as shown in Figure 28. The real-time chart can also be configured to visualize available sensor data in a bar chart plot. The IoT service at Bluemix also shows the information related to a particular device (i.e. connection information, recent events, sensors information and the time it is received) along with the diagnostic log, error codes and connection logs.

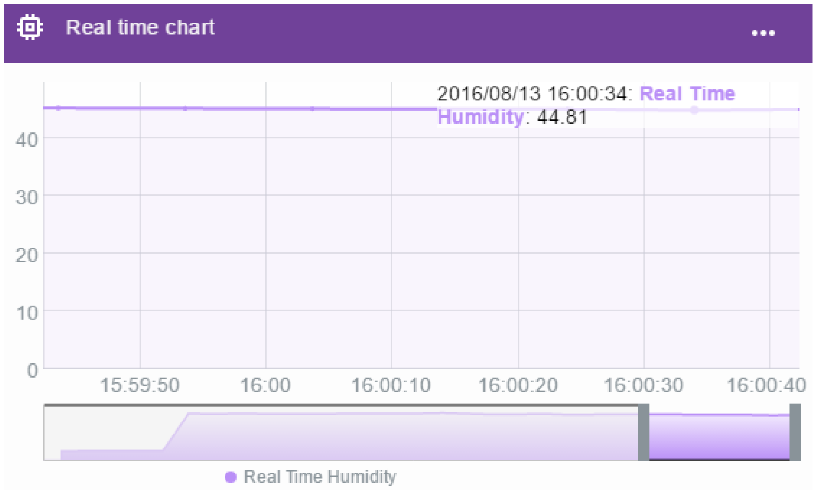


Figure 28: Real time chart in IBM Bluemix shows live humidity sensor data

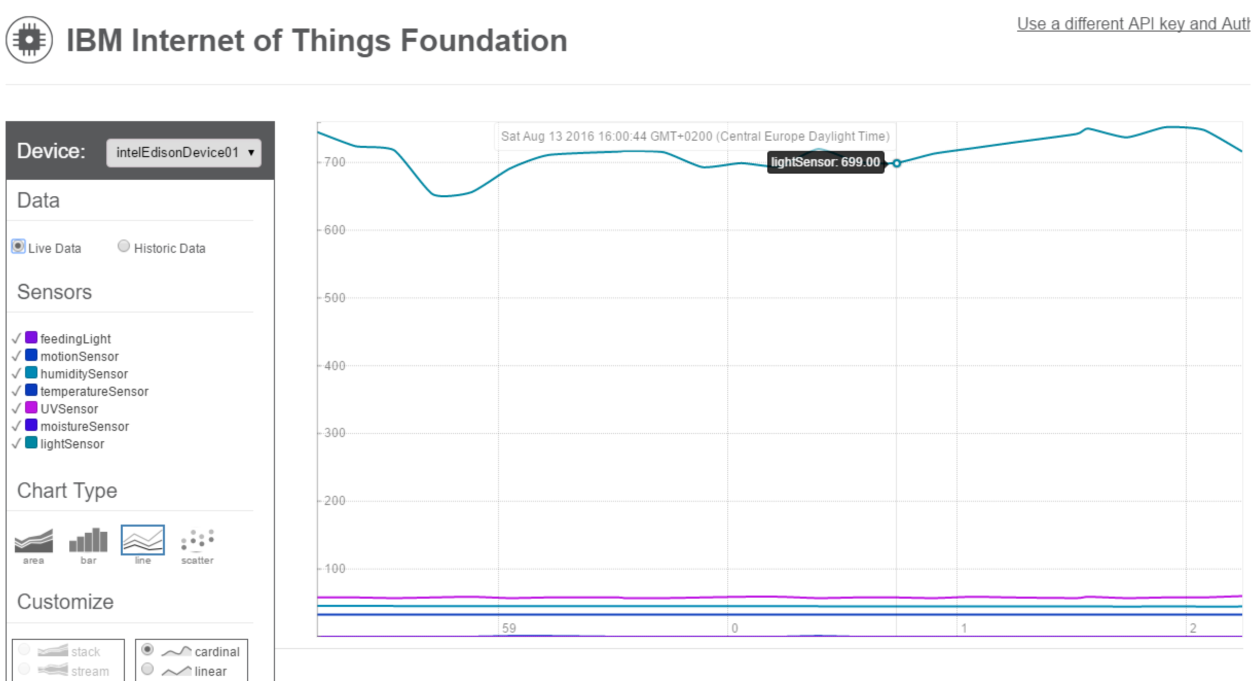


Figure 29: IBM IoT-Visualization (IBM, 2016-c) application shows sensor data

In addition to the in-build application, the IoT-visualization application (IBM, 2016-c), available on GitHub, can be installed on Bluemix that provides more customized options to visualize the sensor data. Figure 29 shows the sensor data by using IoT-visualization application (IBM, 2016-c). The real-time data is coming from the IoT device registered at the IBM Bluemix IoT platform.

4.4.3 IBM Watson IoT for Bluemix Case Study

This section discusses some of the use cases of IBM's IoT platform in different application areas such as weather forecast, transportation and asset management.

Use case 1: The Weather company, now an IBM business, uses the cloud-based Bluemix platform to integrate real-time weather data into the Bluemix application (yadav, 2016). The real-time weather information can be collected from billions of IoT sensors around the world using IBM's IoT solution, and can be used to forecast and detect disruptive weather events (yadav, 2016). The insight from the weather data will be useful for the Bluemix application users to take the appropriate actions based on the analysis, for example, it will be useful for a retailer to know the weather conditions in advance so that he/she can predict a likely increase/decrease in the sale of a particular item.

Use case 2: Nairobi's living roads project provides a cognitive transportation data hub that collects the data from the smartphone sensors (i.e. accelerometers, magnetometers, gyroscopes and GPS) to understand the road conditions across the city (Bryant, 2015). A custom smartphone app known as IBM SteetSense is used to collect the sensor information, IBM InfoSphere Stream is used for real-time analytics, IBM Cloudant database is used to store the data, and IBM Bluemix platform, finally, delivers the results of the analysis. The results are used by the authorities to plan the road maintenance and traffic management (Bryant, 2015).

Use case 3: IBM Maximo software (IBM, n.d.-c) provides asset management (e.g. track and manage asset and location data), work management (e.g. manage planned and unplanned activities), service management, contract management, inventory management and procurement management. By combining IBM Watson IoT with IBM Maximo software can help the asset-intensive industries (i.e. oil and gas, manufacturing, healthcare, transportation) in managing their assets (IBM, 2016-d), for example, it can be used in healthcare to track and locate critical assets, and report the patient data to the health information system. Together with the IoT data, and Watson cognitive feature, IBM Maximo provides the capabilities to meet organisation's asset management needs and helps them in making smarter decisions about their assets.

The existing use cases explain that the IBM Watson IoT for Bluemix can be used to develop and deliver effective IoT solutions in different application areas.

As a part of this research work, the IBM Watson IoT for Bluemix has been evaluated by developing and analysing a use case which is discussed in the next chapter.

Chapter 5. Evaluation of IBM Watson IoT for Bluemix

According to Vaishnavi and Kuchler Jr. (2008), development of artifacts is a major design part in design science research approach. The software artifacts (i.e. web-based sensor monitor application, IoT device control application, and cow tracking application) developed using IBM Watson IoT for Bluemix, are presented in this chapter. It was further suggested by Vaishnavi and Kuchler Jr. (2008) that the evaluation of artifacts can be done by action research, controlled experiments, simulations or scenarios. Therefore, this chapter presents a use case (cow farmhouse) to evaluate IBM Watson IoT for Bluemix. The next section discusses the introduction and motivation for the use case. The system design, hardware and software requirements, implementation using IBM Watson IoT is also discussed later in this chapter.

5.1 Cow farming - Introduction and Motivation

In many countries including Germany, animals such as cows, pigs and chickens are being used as a source of food and/or as a food itself and animal farming has been growing as a business (Scientific Advisory Board on Agricultural Policy, 2015). In this business, the animals are usually kept in a farmhouse which should have all the necessary facilities for the healthy living of the animals. The climate conditions inside the barn where animals spend most of their time can affect the behaviour and health of the animals (GEA, 2015). The main focus of this research is to provide a system that can monitor and control the environment condition inside the barn. The research work is also extended to monitor the activities of the animals using the smart mobile devices. An alternative to tracking the movement of animals can be to use the RFID technique which is already being used in the retail sector to track the movement of goods.

Animal husbandry is the science of managing and caring for animals where animals are being raised for milk, meat, and eggs etc. (merriam-webster). For Germany's agricultural and food industry, animal husbandry is of a major economic importance (Scientific Advisory Board on Agricultural Policy, 2015). Also, it has considerable relevance for the development of value chain, jobs, etc. in rural areas (Scientific Advisory Board on Agricultural Policy, 2015). In the European Union, Germany is one of the four largest producers in the farming sectors (Federal Ministry of Food and Agriculture, 2014), and no other country in Europe produces milk or pork more than that is being produced by Germany.

As part of this research, an example of 'cow farming' is presented. The well-being of animals living in the farms is of significant importance. In Germany three out of four cows live in loafing sheds (Federal Ministry of Food and Agriculture, 2014), and also there is a regulation for providing the animals with outdoor climate. The barn climate is affected by a change in the temperature, light, wind, humidity, animal-related thermal processes within the barn (GEA, 2015) etc. Monitoring and controlling these factors is an essential requirement for the good health of the animals living in the barn. The concept of IoT can be used to implement a system that can monitor the barn climate and control the devices from remote end. The IoT cloud-based platforms can be used to collect and store the real-time sensor data that can later be used to get the insight from the data, such as if the temperature was adequate in the last few days or not. The tracking functionality can help in tracking the cow movements. The software and hardware required for the implementation are discussed in next sections.

5.2 System Design

This section explains the design of the use case in detail. The use case is divided into two parts. The first part explains the barn climate monitor and control system, and the second part is about the cow tracking system. The IoT-A modelling approach as discussed in section 3.4 is used to design the system. First, the typical barn structure and modelling approach is introduced, and later hardware/software requirements and the implementation of the systems using IBM Watson IoT for Bluemix, Grove indoor environment kit, and smart mobile devices are discussed in detail. As mentioned in the design science research (Chapter 2), the use case is implemented to evaluate the IBM's IoT platform. It will help in understanding the capabilities of IBM's IoT platform.

5.2.1 Typical Barn Layout

The cow farmhouse is divided into four major areas as shown in Figure 30. The device inside the barn including cooling system, heating system, light bulbs etc. can be controlled using the web/mobile based application discussed later in this research report.

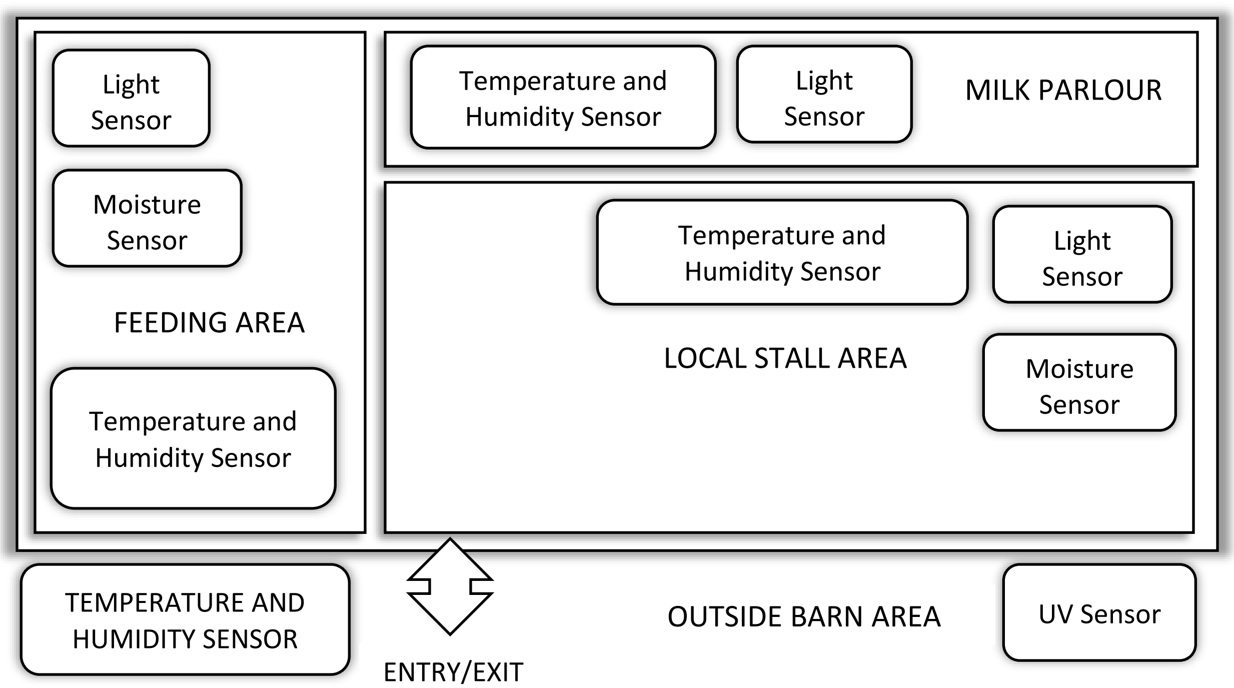


Figure 30: A typical barn layout, adapted from (Rockey, 1965, p. 3), and sensor's positioning

The barn area and arrangement of sensors in these areas are discussed below.

- **Local stall area:** In this area, cows take a rest and sleep. This area will have a temperature and humidity sensor, moisture sensor to measure the moisture in the soil where the cows will spend most of their time, and a light sensor to measure the light intensity in this area.
- **Milk parlour:** It is an area for automatic milking facilities. This area will have a temperature and humidity sensor and a light sensor.
- **Feeding area:** A special area for feeding the cows. The feeding area will have a temperature and humidity sensor, a moisture sensor to measure the moisture in the food (e.g. moisture in grass), and a light sensor.
- **Outside the barn** (But, inside the farm area): This area will have an UV sensor to measure the Ultra-violet radiation level outside the barn so as to check if it is suitable for the animals to go out, and a temperature and humidity sensor to measure the current temperature and humidity level outside the barn.

The user can monitor the data generated from the sensors through a web/mobile based application. The sensors will send real-time data to the web/mobile application servers. Web/Mobile application servers will collect the data from these sensors and analyse the data through the application(s) designed for this kind of information. The web/mobile application will control the smart devices on the basis of different conditions and correspond defined logics to handle such conditions.

5.2.2 Domain Model and Information View

Many people believe that graphical techniques will dominate the software development process in the future (Fowler, 2004). Figure 31 shows the **Domain Model** (DM) that explains the concepts and the relationships between the concepts. The blue colour box identifies a hardware component, the green colour is used to identify a software, yellow represents the animate being (i.e. Human being, animals), and the brownish coloured box is used when the category is not defined or the concept has multiple categories. The DM approach uses the UML notations.

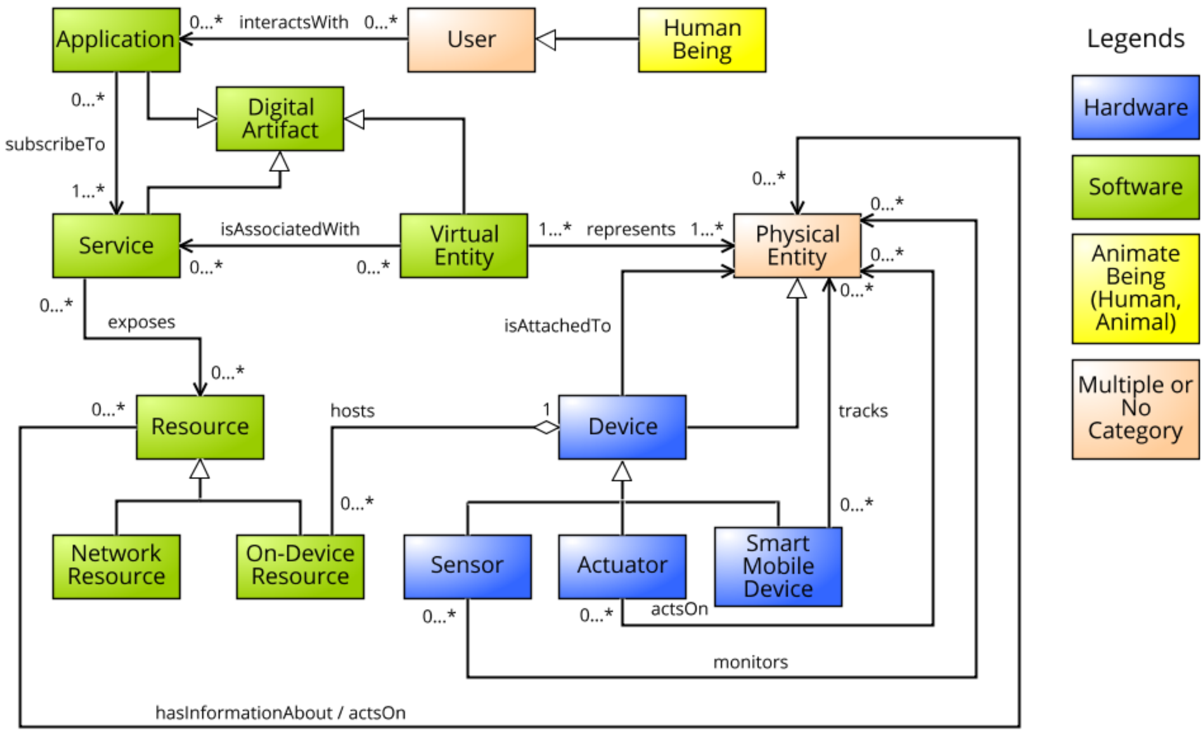


Figure 31: IoT-A (IoT-A, 2013) Domain Model – Barn climate monitoring, controlling, and Cow tracking, own illustration

As shown in figure above, the sensors, actuators, and smart mobile devices can be represented as a device (using UML generalisation). A sensor can monitor the physical entity (barn climate), an actuator can act on other physical entities (e.g. turn on the light system), and a smart mobile device can track the movement of a physical entity (e.g. cows). All the physical entities can be represented by a virtual entity in the software. A resource represents a software that is used to communicate between the IoT device and IoT platform. An IoT platform offers different services (e.g. databases, analytics) to communicate with the remote IoT devices and handle the data generated by them. An authorised human being can access the IoT services through the web/mobile based applications.

The **Information View** (Figure 32) gives the layered view of the system and shows the different components in these layers.

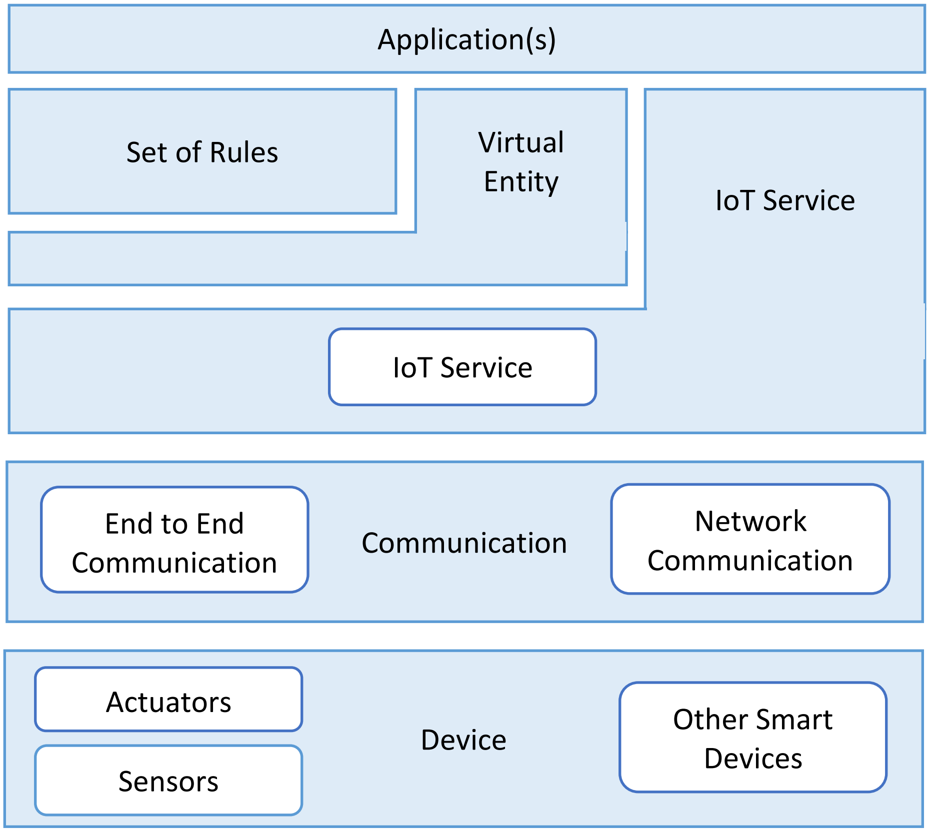


Figure 32: IoT-A (IoT-A, 2013) Information View - Cow farmhouse system, own illustration

The lower layer shows the physical devices. A device can communicate with the IoT services over the internet. An application can directly communicate with the IoT service. We can have a set of rules to control the virtual entities. A web-based or mobile-based application (shown as upper layer) can be used to remotely monitor the sensor data or control the IoT devices.

The system is divided into two parts 'Barn environment monitor and control' and 'cow tracking' respectively and is explained in detail in next two sections.

5.2.3 Part A – Barn Environment Monitoring and Controlling

This part discusses the system that monitors the environment conditions in different parts of the barn using the sensor devices deployed in the farmhouse and IBM's IoT solution. The system also allows the user to control the devices in the specific area (e.g. light system in local barn area).

5.2.3.1 Hardware and Software Requirement

The following hardware and software are used to implement the barn environment monitor and control system.

- **Grove indoor environment kit**

Grove indoor environment kit is a functional kit that consists of a set of different sensors that makes it possible to create indoor environment applications with the help of Intel Edison board and Intel compute module. This kit contains the following modules. Some of them are used to implement the system.

- **Base shield V2**: It is an expansion board. Base shield v2 has many Grove connectors, making it convenient to use Grove products together.
- **Temperature and humidity sensor:** This is a multifunctional sensor that gives temperature and relative humidity information at the same time. It provides reliable readings when environment humidity condition is in-between 0-80% RH, and temperature condition is in-between 0-70°C, covering needs in a most home and daily applications that don't contain extreme conditions.
- **Moisture sensor:** This moisture sensor can be used to detect the moisture of soil or judge if there is water around the sensor.
- **Light sensor:** The light sensor module uses the photo-resistor to detect the light intensity of the environment. The resistance of the sensor decreases when the light intensity of the environment increases.
- **UV sensor:** The Grove - UV Sensor is used for detecting the intensity of incident ultraviolet (UV) radiation. The module will output electrical signal which is varied with the change of the UV intensity. UV sensors are used for determining exposure to ultraviolet radiation in the laboratory or environmental settings.
- **PIR motion sensor**: This is a simple to use PIR motion sensor with Grove compatible interface.
- **Encoder:** This module is an incremental rotary encoder. It encodes the rotation signal from the axis and output the signal by the electronic pulse.
- **Button (P)** (P stands for 'panel mount'): This is identical to the original Groove button, except the Grove connector is moved to the back so that you can easily use it as a neat and wire-free human interface device.
- **LCD RGB backlight:** This Grove enables you to set the colour to whatever you like via the simple and concise Grove interface.
- **Relay:** Relay is a digital normally open switch that controls a relay capable of switching much higher voltages and currents than the normal Arduino boards. When set to HIGH, the LED will light up and the relay will close allowing current to flow.
- **Servo:** It is available as plug and play.
- **Buzzer:** The piezo can be connected to digital outputs, and will emit a tone when the output is high. Alternatively, it can be connected to an Analogue pulse-width modulation output to generate various tones and effects.)
- **9V to Barrel Jack Adapter:** This adapter consists of a 9V battery clip and a standard 5.5*2.1mm center positive barrel jack, it applied to power any device needs 9V input voltage and has on-board barrel jack.
- **26AWG Cables**
- **USB Cable**

- **Intel® Edison kit for Arduino and compute module**

The Intel® Edison module is a SoC (System on Chip) that includes an Intel® Atom™ 500MHz dual-core, dual-threaded CPU and an Intel® Quark™ 100 MHz microcontroller. It has integrated Wi-Fi, Bluetooth 4.0 LE, and support for Yocto Linux, Python, Node.js and Wolfram. Intel Edison expansion board has support for connecting sensors to analogue, digital, or I2C (Inter-integrated Circuit) interface.

- **Intel XDK IoT IDE**

It is an Integrated Development Environment (IDE) that enables you to create the application for the Intel IoT boards. The IDE software is available for Windows, Linus, and Mac OS.

- **IBM Watson IoT for Bluemix**

As discussed in sections 4.1.1 and 4.3, IBM Bluemix is a cloud-based platform that has multiple programming languages support (e.g. Java, Node.js, Go, PHP, Python, etc.) to implement the web-based application.

5.2.3.2 Connecting Grove Kit, Intel Edison module and IBM Bluemix IoT Platform

Figure 33 shows the basic connectivity of different hardware and software components of the barn climate monitor and control system.

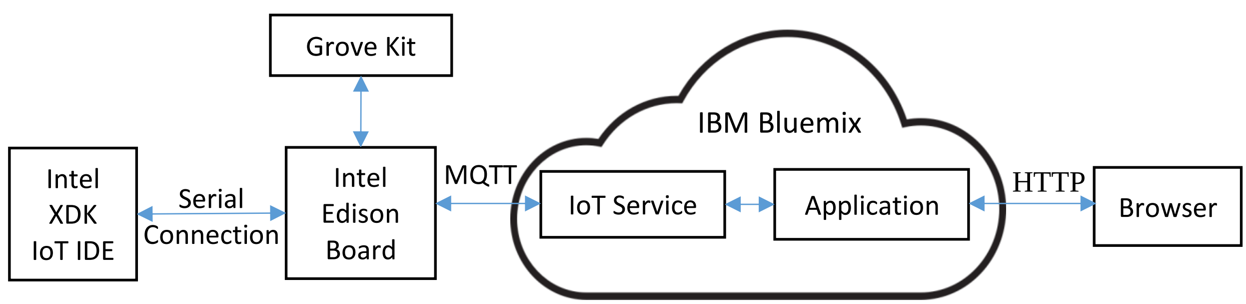


Figure 33: Barn climate monitor system components and connectivity, own illustration

IBM Bluemix IoT platform supports MQTT protocol (section 3.6.5) to communicate with the smart devices. Grove sensors can be attached to the Intel Edison board using the base shield as shown in Figure 34. Table 12 shows the Grove module connection to the analogue (A), digital (D) and I2C pins of the Intel Edison board. For this research work, we have not used the actual barn environment. The actual sensors and other IoT devices are used in the laboratory environment, and it is expected to work in the same way in the real scenario.

Figure 34: Grove indoor environment kit components on Intel Edison board, adapted from (Seeed, 2015)

Grove Module	Connected to Edison board pin
Temperature and Humidity sensor	I2C
Moisture sensor	A1
Light sensor	A2
UV sensor	A3
PIR motion sensor	D7
Encoder	D2
Button	UART (D1)
LCD RGB Backlight	I2C
Relay	D5
Servo	D6
Buzzer	D4

Table 12: Grove module connectivity on Intel Edison board, adapted from (Seeed, 2015)

After physically connecting all the hardware components, the executable generated using Intel XDK IoT IDE, which receives the sensors data and controls the actuators, is deployed to the Intel Edison compute module. The next section discusses the implementations at the IBM Bluemix IoT platform to handle the data coming from the sensors and sending commands to the actuators.

5.2.3.3 Implementation using IBM Bluemix

The Node.js application is used to implement the barn climate monitoring and barn device controlling system as shown in Figure 35 and Figure 36 respectively. The Bluemix IoT service sends the sensors data to the Node.js application which receives it on the ibmiot input node (FeedingAreaSensor node shown in blue colour in Figure 35). The function nodes of Node.js applications parse the sensor data and send it to the web application over the preconfigured Web-Socket connection. A detailed flow is discussed in the next section.

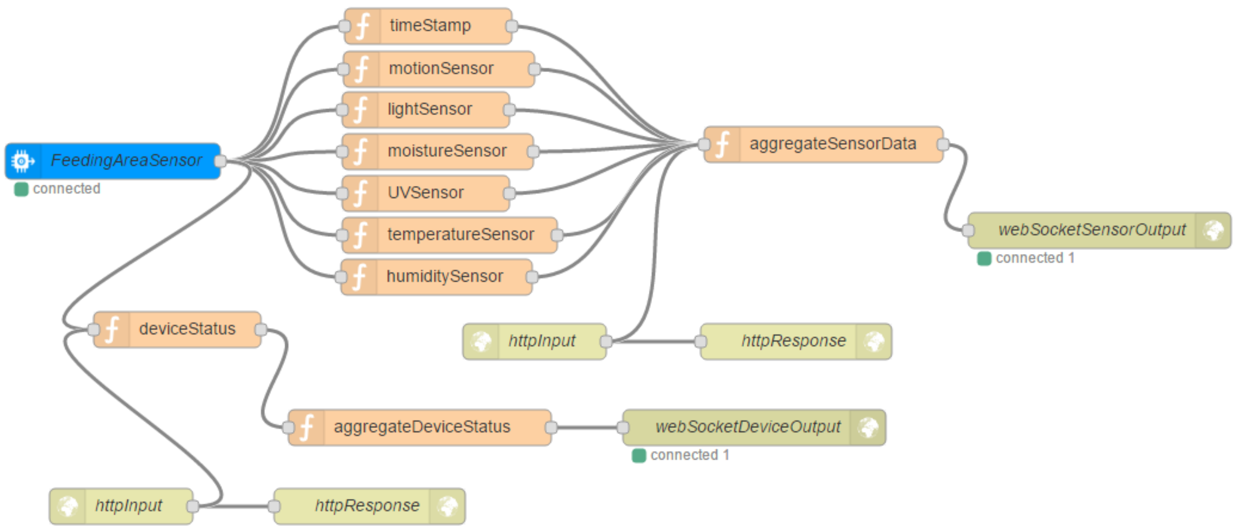


Figure 35: Node.js flow for monitoring barn climate

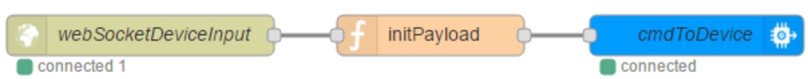


Figure 36: Node.js flow for controlling barn devices

To send the command to the IoT device connected to the Bluemix IoT platform, web-based application sends the input to the Node.js application which construct the message in JSON format and sends it to the Bluemix IoT service through the ibmiot output node (cmdToDevice node shown in Figure 36).

The next two consecutive sections discuss the interaction between different components in detail.

5.2.3.4 Interactions – Monitor Barn Climate

This section explains the interaction between the different components to monitor the barn climate. The components and their interaction are shown in Figure 37. In the following flow, it is assumed that the devices are registered with the Bluemix IoT platform and all the basic physical connectivity is already established.

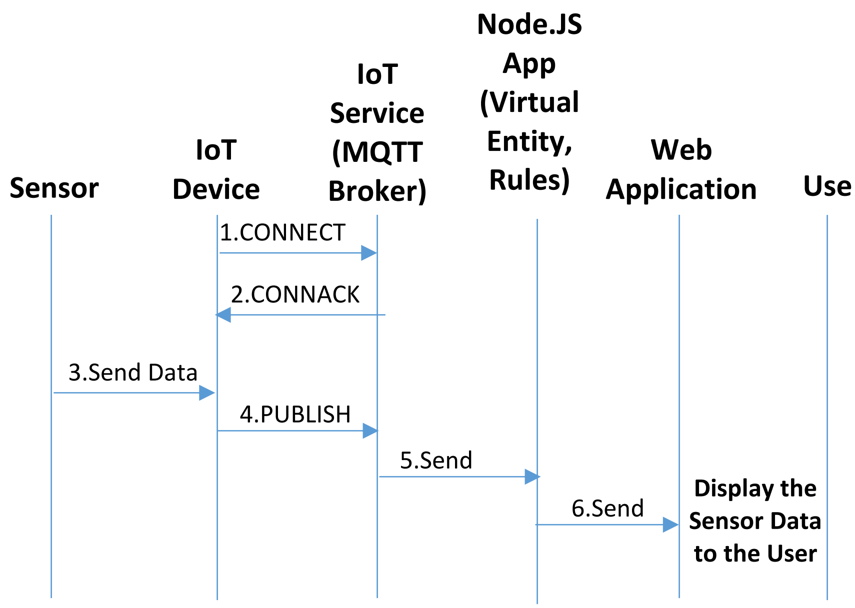


Figure 37: Interactions - Monitor barn climate

1. The IoT device (Intel Edison compute module) sends the MQTT connect message to the IoT service (MQTT Broker) running at the Bluemix platform.
2. The IoT service replies with acknowledgement message if the connection is successful.
3. In this step, the actual sensor devices (e.g. temperature sensor) will generate the data.
4. The data generated by the sensors is received by the Intel Edison module (IoT device) and published to the IBM Bluemix IoT platform. The MQTT publish message has the same topic that is subscribed in step 3 by the node.js application.
5. The MQTT publish is sent to the node.js application's MQTT receive module.
6. Node.js application has the predefined rules to handle the sensor data. The node.js application will categorise all the sensor data on the basis of the event defined in the topic field and sends it to the browser application where this real-time data can be monitored by the human user.

5.2.3.5 Result – Barn Climate Monitor

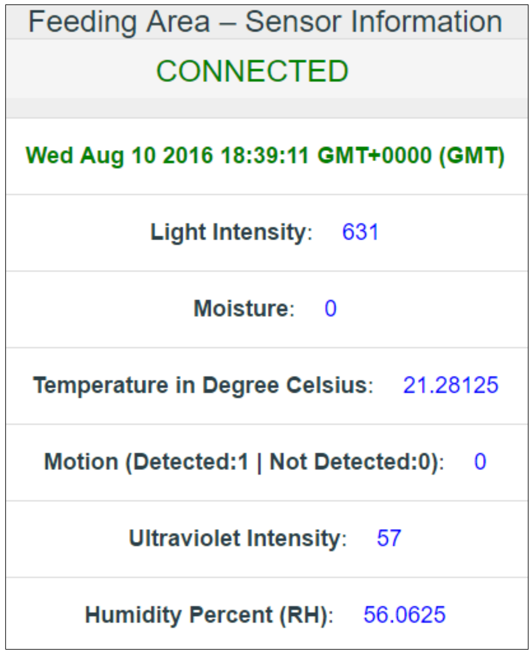


Figure 38: Web-based application displaying the real-time sensor data

Figure 38 show the real-time sensor data captured by the web-based application.

5.2.3.6 Interactions – Control Barn Devices

Along with monitoring the sensors data (section 5.2.3.4) we can also control the devices (e.g. light system) located at the specific area in the barn. Figure 39 shows the interaction between different components for controlling these devices. Again, it is assumed that the devices are registered with the Bluemix IoT platform and all the basic connectivity is already established.

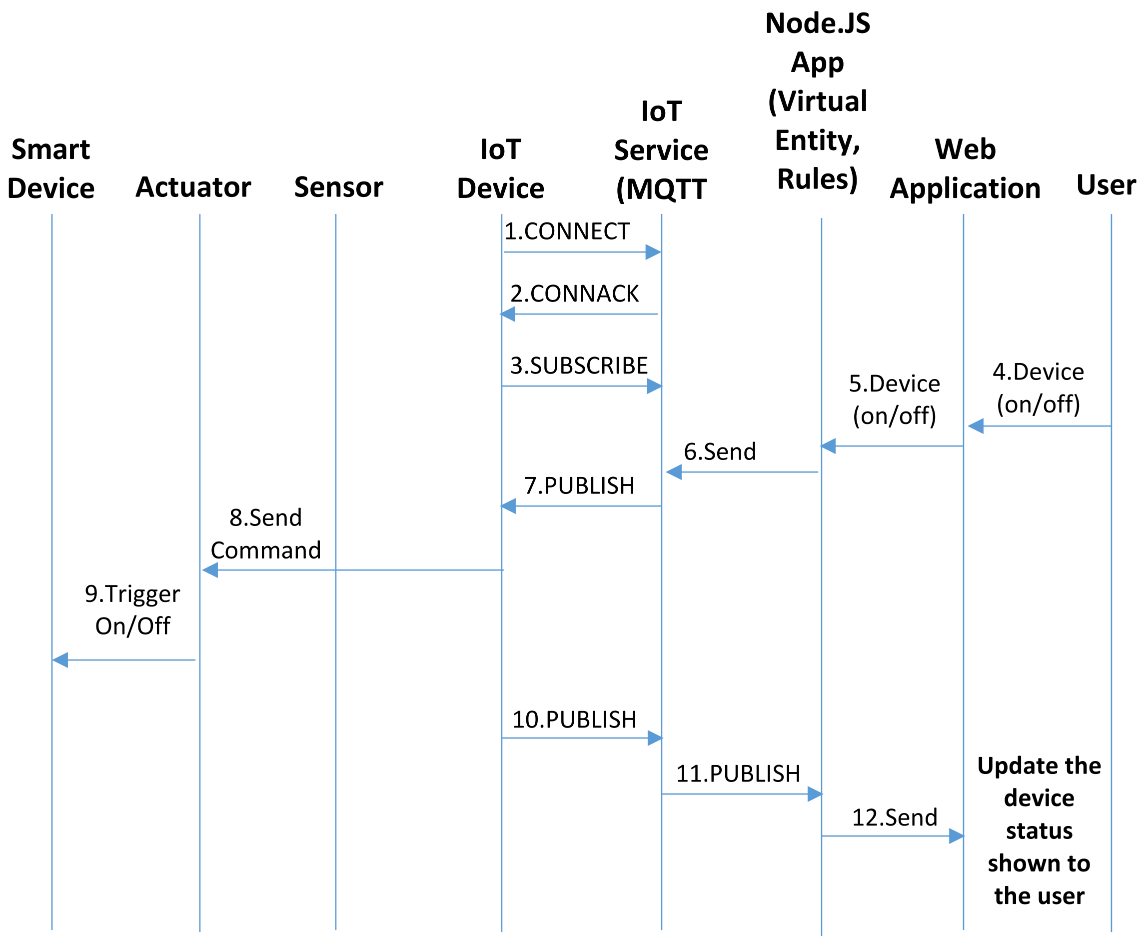


Figure 39: Interactions - Barn device control

1. The IoT device (Intel Edison compute module) sends the MQTT connect message to the IoT service (MQTT Broker) running at the Bluemix platform.
2. The IoT service replies with acknowledgement message if the connection is successful.
3. The IoT device subscribed to topic that instruct it to control the attached devices.
4. User press the button to turn on/off the device.
5. Rules are defined at the Node.js application that acts on the data received from the web application over the Web-Socket connection between the web application (e.g. PHP application) and the Node.js application.
6. The Node.js application converts the user input into the MQTT command and sends it to the Bluemix IoT service.
7. The Bluemix IoT service publishes the command to the IoT device.
8. The IoT device instruct the actuator to take action based on the command.
9. The actuator turns on/off the device based on the command/instruction provided to it.
10. The IoT device publishes the current status of the device (e.g. light on/off).
11. The Node.js application receives the current device status and send it to the application.

5.2.3.7 Results – Barn Device Control

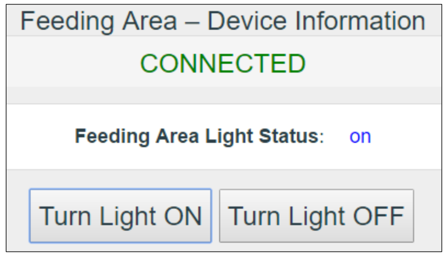


Figure 40: Web-based application displaying status of device and options to on/off the device

Figure 40 shows the snap-shot of the web-based application that displays the current status of the device located at the barn and also provides the options to turn on or off the device.

5.2.4 Part B – Cow Tracking System

The second part discusses the part of the system that can track the cow movements. Two methods are proposed to achieve this.

In the first method, each cow will be identified by a Tag-ID which stores the basic information about the cow (i.e. Breed, Gender, Age, etc.). In each barn area (e.g. feeding area, milk parlour, etc.), RFID detectors can be used for these tags. These detectors will send the real-time data to the IBM Bluemix IoT server. With the help of this data, it is possible to analyse in which area (e.g. feeding, milk parlour, etc.) a particular cow spends most of its time. This method is not implemented in this research work and only mentioned here to discuss the different possibilities to implement the cow tracking system.

The second method uses the smart mobile device, where one mobile device can be attached to one cow. An already available mobile application from IBM, implemented by Beguelin (2015), is used to send the location data to the IBM Bluemix. The next section discusses the method to track the movement using mobile device and IBM Watson IoT in detail.

5.2.4.1 Hardware and Software Requirements

The following hardware and software are used to implement the cow tracking system (using android devices to track the movement).

- **Android mobile device**

Android based mobile device with in-built GPS system is used to track the movement of a cow.

- **IBM Watson IoT for Bluemix**

Refer section 4.4.

- **Google Map API**

The Google Map API is used to display the location of the moving object (e.g. cow) on the Google map. To use the Google Map API, we need a Google user account.

- **Android studio**

The Android studio can be used to build the applications for the android devices.

5.2.4.2 Implementation using IBM Bluemix

The Node.js application implemented in IBM Bluemix to track the movement of cows is shown in Figure 41. The ibmiot input node (locationData) receives the sensor data from IBM IoT service. The function nodes parse the data and send it to the web-based application over a preconfigured Web-Socket connection. A detailed flow is discussed in next section.

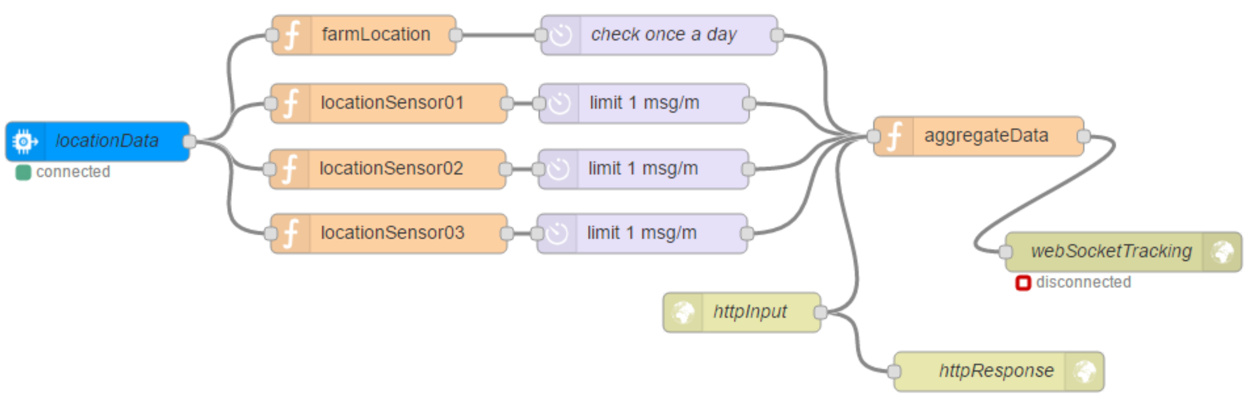


Figure 41: Node.js flow for tracking cow movement

5.2.4.3 Interactions – Cow Tracking System

Figure 42 shows the interaction between different components to track the movement of each cow.

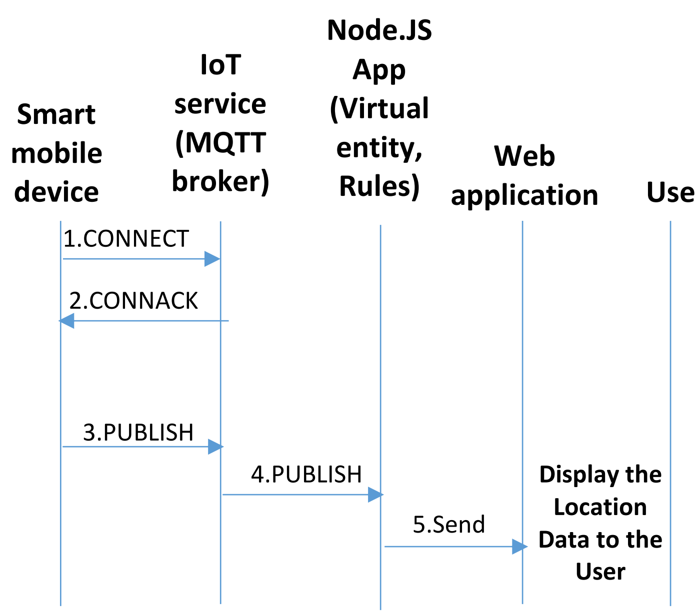


Figure 42: Interactions - Cow tracking system

1. The IoT device (Intel Edison compute module) sends the MQTT connect message to the IoT service (MQTT Broker) running at the Bluemix platform.
2. The IoT service replies with acknowledgement message if the connection is successful.
3. The smart mobile device gathers its GPS coordinates (e.g. longitude and latitude) and publish it to the IBM Bluemix IoT platform's MQTT broker.
4. MQTT broker publishes the message to the node.js application's MQTT input module.
5. The MQTT message is parsed at the node.js application and location information of each cow is send to the web application over a preconfigured web socket. And, finally, the data is visible to the user. The web application is implemented in PHP using the IBM Bluemix runtime environment and Google Map API.

5.2.4.4 Results – Cow tracking system

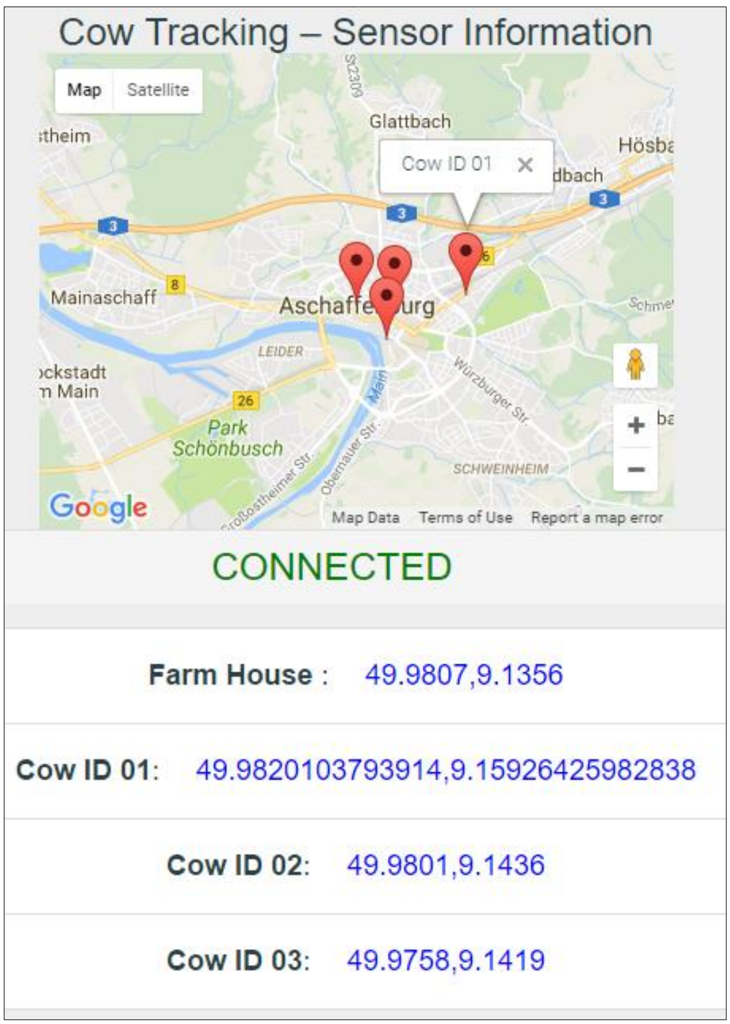


Figure 43: Web-based application displaying the location of cow

Figure 43 shows the web-based application snap-shot. The Google Maps application shows the real-time location of the cows and the text information shows the coordinates of the location.

The application allows to click on the coordinates corresponding to each cow to show the exact location of that cow on the Google map in separate browser tab.

5.2.5 Summary/Comment

The IBM Watson IoT for Bluemix is evaluated by implementing a use case. The Internet of Things service of Bluemix provides a scalable solution to connect multiple IoT devices. The communication between the IoT device and the Bluemix IoT service is done over MQTT protocol which is a lightweight protocol suitable for small IoT devices. The Bluemix platform provides the runtime environment that can be used to develop web-based and mobile-based applications in different programming languages. The Bluemix provides the possibilities to store the source code in Built-in Jazz source control, Bluemix's hosted Git repository, and external GitHub. The online editor and pipeline concept of Build-Test-Deploy makes it easy to develop, test and deploy the applications using the Bluemix platform. For the use case, the Node.js runtime environment was used to create the interface between the web-based application and Bluemix's IoT service. The visualization of sensor data is also possible using the already existing service in Bluemix. Many online recipes are also available that explains the ways to connect IBM IoT solution with the existing hardware.

Overall, the IBM Watson IoT for Bluemix is a good IoT offering by IBM. As shown in the use case scenario (see Chapter 5), IBM Watson IoT provides a scalable solution to register and connect multiple IoT devices. Using IBM Bluemix, different storage technologies (Cloudant NoSQL DB, dashDB, IBM DB2 on cloud, MongoDB by Compose, SQL Database) can be used as per the requirements and priority which makes it a very flexible solution. Also, the pipeline concept (see Figure 25), storage, built-in repository, use of external Git, multiple programming language support etc. makes Watson IoT a good developer friendly platform. Watson IoT offering such as analytics, risk management, connect and information management, as shown in Figure 26, makes it a complete IoT solution. The Watson cognitive feature can be used to get the insight from the real-time or historical IoT data.

Some challenges in IoT still need to be addressed by the existing IoT development platform vendors such as standardisation and interoperability. The different IoT communication protocols are used by the IoT development platform providers. There is a need to have consensus on the way to handle IoT devices so that a globally accepted IoT solution can be delivered.

The next chapter concludes the thesis and gives an overview of the knowledge gained from this research work.

Chapter 6. Conclusion

This chapter concludes the thesis and gives the direction to the future work in the IoT domain.

The design science research methodology was used to design the research steps that guide the achievement of the research objectives, and to answer the related research questions. The design science research steps, i.e. awareness of problems and suggestions, were used to gain the theoretical knowledge and understand the problem domain. Various literature resources have been studied to understand the concepts, challenges, standards and technology in the field of IoT research and development. The existing cloud-based IoT development platforms are examined and compared to visualize their functionalities. As a part of design science approach, the software artifacts (IoT applications) were developed and evaluated in this research. A use case, discussed in Chapter 5, is implemented to evaluate the IBM Bluemix's IoT solution. This section presents the final design science research step (i.e. conclusion of research).

The main objective of this research was to compare the capabilities of various cloud-based IoT development platforms and to analyse one platform for in-depth evaluation by implementing a use case scenario. General, business and technical features of the five commercially used and popular IoT platforms IBM Watson IoT for Bluemix, Microsoft Azure IoT suite, Amazon AWS IoT, Google Cloud Platform for IoT, and ThingWorx were compared. All the examined platforms have delivered a promising IoT development solutions. A detailed comparison table, as shown in section 4.2, shows the similarities and differences between the examined platforms.

The IOT-A modelling approach has been used to explain the domain in detail. The domain model approach shows the different components and their relationships and provides a common vocabulary that is independent of any programming language.

Four research objectives (see section 1.3) were identified and addressed in this research.

The first research objective was to understand the concept of IoT and related challenges. In general, IoT can be defined as a network of sensors, actuators, IoT devices and clouds, connected through the Internet technology. The role of cloud computing is very important. With the efficient use of cloud approach, the IoT can be spread globally and provide more opportunities from the point of view of business, technology and society. There is still an uncertainty about the IoT because there is no globally accepted definition of IoT. The uncertainty also grows due to lack of common standards for communication between the different components of IoT. The various standard bodies are working in the direction to shape the IoT as mentioned in section 3.2, but, still, they need to work together. The IoT ecosystem is hard to define because of the rapid development in this area. The key players as mentioned by IEEE-SA ecosystem study are market, technology and standards.

The second research objective was to compare different existing cloud-based IoT development platforms so as to understand the capabilities they have, and to find out the similarity and differences between these platforms. The comparison is discussed in Table 9 and Table 10 on the basis of general, business and technical criteria mentioned in Table 7 and Table 8. The comparison shows that all the platforms have provided the basic IoT functionalities such as connect IoT devices and application to the cloud, store and visualize IoT data. While ThingWorks, a model-based platform, uses cloud environment of Amazon and Microsoft, all other platforms are using their own cloud system. The comparison of IoT development platforms is summarised in section 4.2.3.

The third research objective was to select one platform from the studied platforms for in-depth analysis, and the fourth objective was to evaluate this selected platform by implementing a use case scenario. On the basis of comparison of different IoT platforms, IBM Watson for Bluemix was selected to understand its IoT offerings. IBM Watson IoT for Bluemix is based on the open source cloud foundry and is more developer friendly because of the built-in runtime environment. A challenging use case was selected and developed using the IBM Bluemix IoT platform. The use case is designed using the IOT-A modelling approach. The use case evaluation shows that developing a web-based application on Bluemix is very easy, and its IoT service provides flexibility and scalability. The integrated Node.js (runtime) environment makes it possible to define a set of rules on the data collected from the sensors and also makes it easy to communicate with the web-based applications through the Web-Socket. The visualization application allows analysing the real-time and historical data. The analysis and evaluation of IBM Watson IoT for Bluemix are summarised in section 5.2.5.

For this research, the comparison of different IoT development platforms was done by the study of available documentations and tutorials from the respective platform vendors. This research work can be extended in future by comparing the existing platforms on the basis of performance, scalability, storage etc. This can be achieved by running simulations on every platform and monitoring the response time, cost per device or message etc.

The overall result of this research work shows that a lot of work has been done in the area of Internet of Things and a wide range of domains, application areas have been benefited from it. But, the issues like social, policy, privacy, standardization, interoperability are still needed to be addressed properly to justify the ITU-T notion of “anything connected”.

Appendix A: IoT Platform's Graphical User Interface

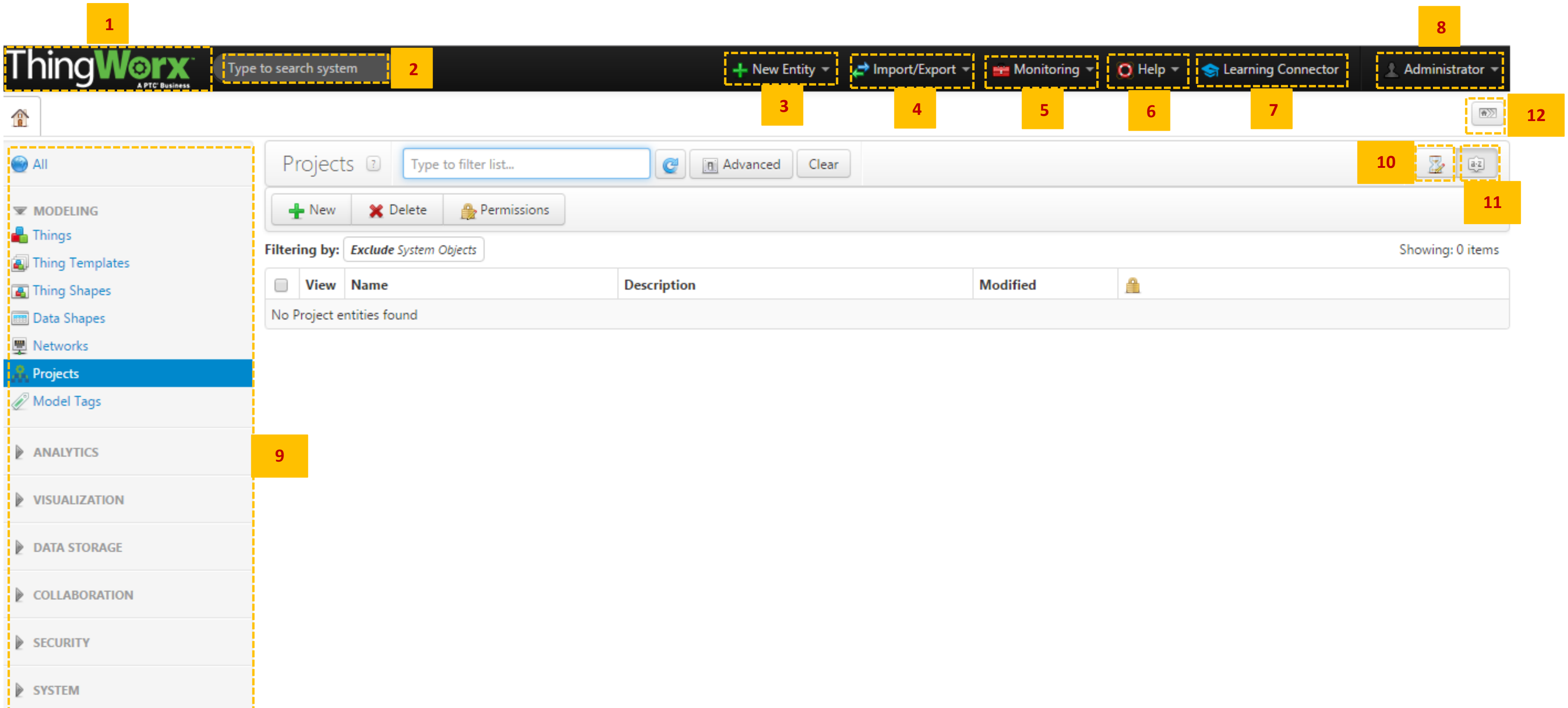
IBM Bluemix Console

	IBM Bluemix console: Description
1	Go to console home.
2	Try new Bluemix console.
3	Go to dashboard home.
4	Discover products and solutions.

	IBM Bluemix console: Description
5	Catalog: Boilerplates (shortcut to start with new application), compute (start with runtimes, container images), services (Watson, mobile, DevOps, web and application, network, integration, data and analytics, security, storage, business analytics, Internet of Things, APIs).
6	Price estimations and schemes.
7	Documentations on using Bluemix platform and services.
8	Link to Bluemix's developer community.
9	Number of days left for subscription to expire.
10	Information related to user's account, status, logout, selecting a region, organization name, manage organization, notifications, support, submit an idea.
11	Grid view on dashboard.
12	List view on dashboard.
13	Search option.
14	Hide side menu.
15	Create a space.
16	Existing space and related services and applications etc.
17	Create Cloud Foundry application.
18	Start using containers.
19	Run virtual servers.
20	Work with data (analytics).
21	Use services or APIs.
22	List of current applications.
23	List of current services.

Table 13: IBM Bluemix console description

ThingWorx Console (Composer)

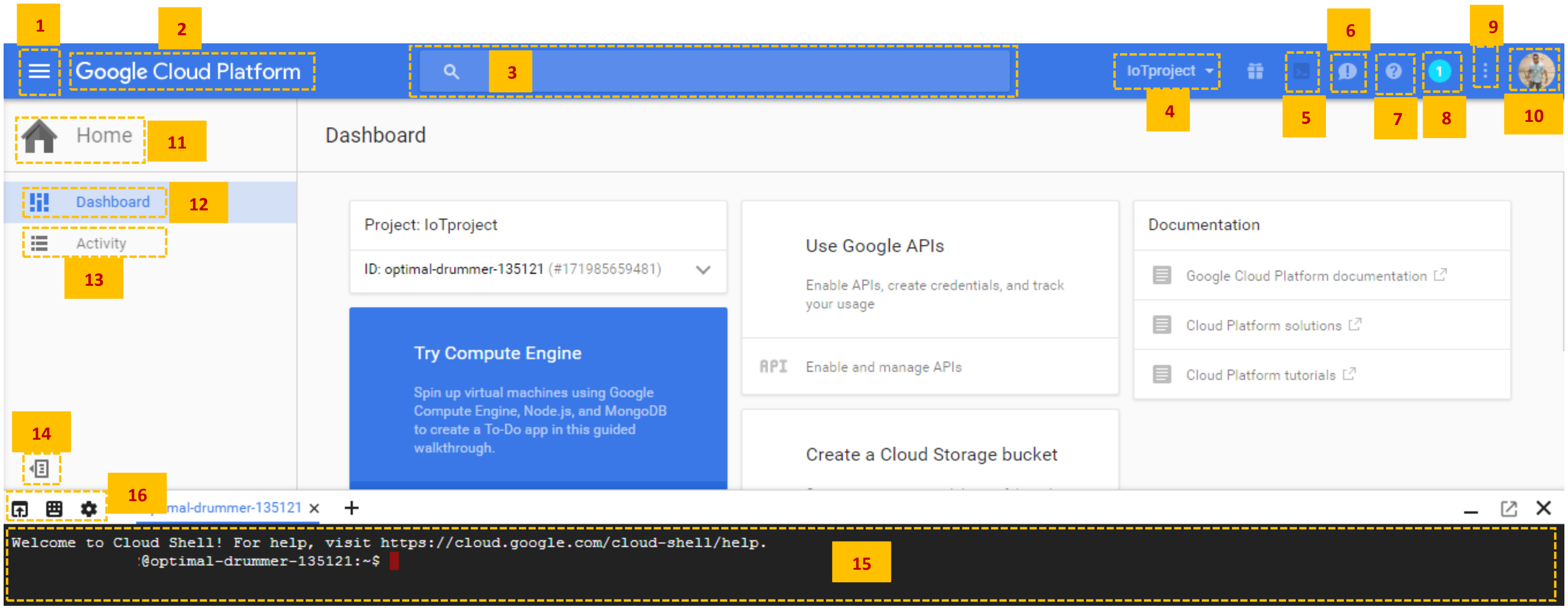


	ThingWorx console (Composer): Description
1	Composer home.
2	Search in the system.
3	Create a new entity from a list of available entities.
4	Import/export from file, ThingWorx storage, source control entities.
5	Monitoring: logs (application, communication, composer, configuration, security, script), active edits of owner or all users, status (connection servers, remote “things”, subsystems, system, users logged in), alerts (alert summary, alert history) etc.
6	Help: documentations, license etc.

	ThingWorx console (Composer): Description
7	Open PTC's learning connectors for tutorials etc.
8	User's login related information: preferences, change password, logout etc.
9	Side menu contains shortcuts for available services (i.e. visualization, data storage etc.)
10	Sort the results by date of modification.
11	Sort the results alphabetically.
12	Close all open tabs.

Table 14: ThingWorx console (Composer) description

Google Cloud Platform Console

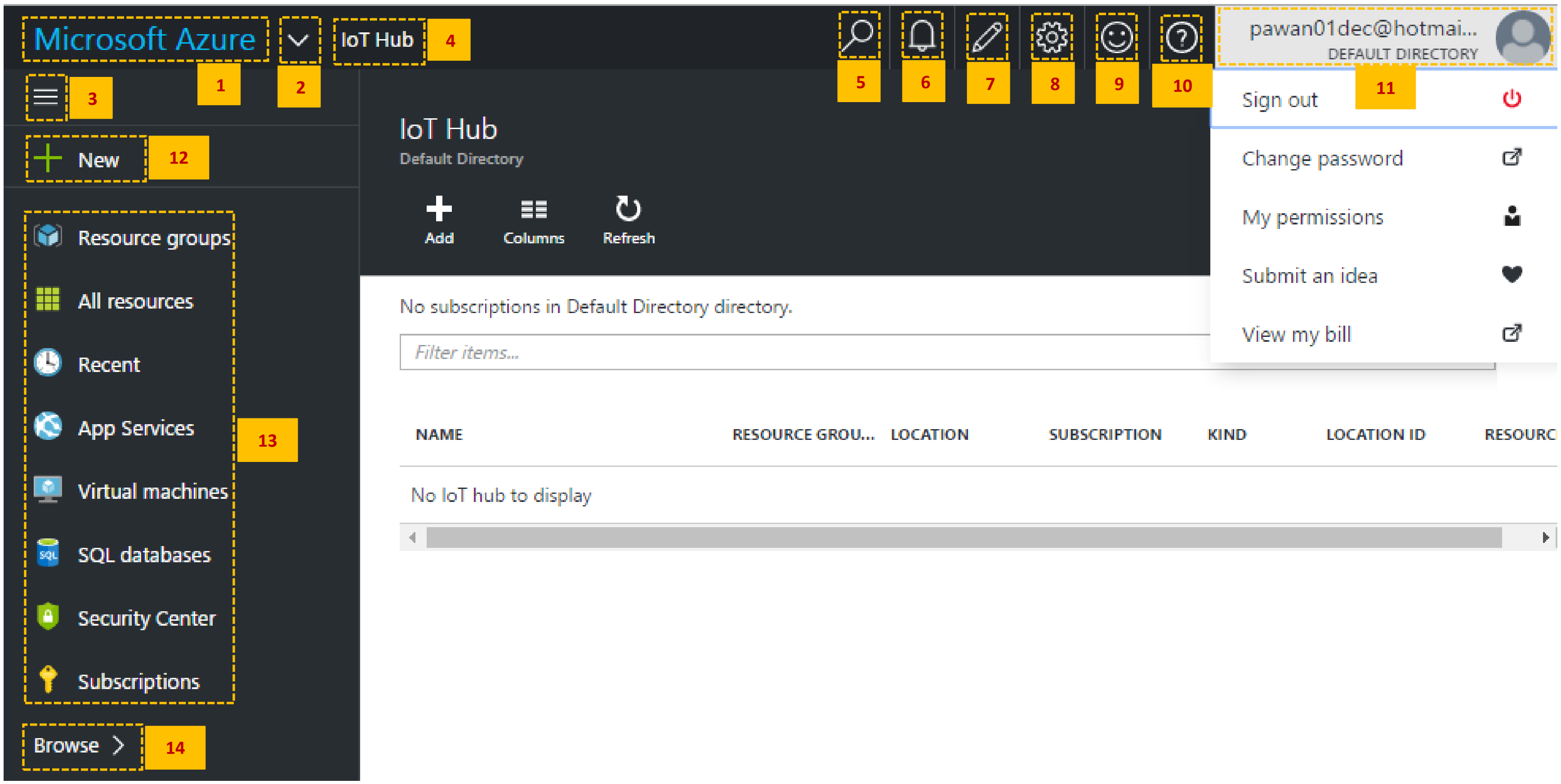


	Google cloud platform console: Description
1	Open menu to get a list of products and services: go to home, API manager, billing, cloud launcher, support, IAM & admin, compute (app engine, compute engine, container engine, networking), storage (Bigtable, SQL, Datastorem storage), stack driver (monitoring, debug, trace, logging, error reporting), tools (development, deployment manager, test lab), big data (BigQuery, Pub/Sub, Dataproc, Dataflow, Genomics).
2	Go to dashboard home.
3	Search product and services.
4	Current opened project. Also, options for creating a new project, manage all projects, etc.
5	Activate Google cloud shell.
6	Send feedback.
7	Console help.

	Google cloud platform console: Description
8	Notifications of activities.
9	Utility and more: language preference, keyboard shortcuts, downloads, support, terms of services, privacy, project information, tour to console, etc.
10	User login information.
11	Console home.
12	Go to dashboard home.
13	Go to all activities (activity log of project or application created by user by date and time).
14	Hide side menu.
15	Cloud shell terminal.
16	Cloud shell settings and other options.

Table 15: Google Cloud platform console description

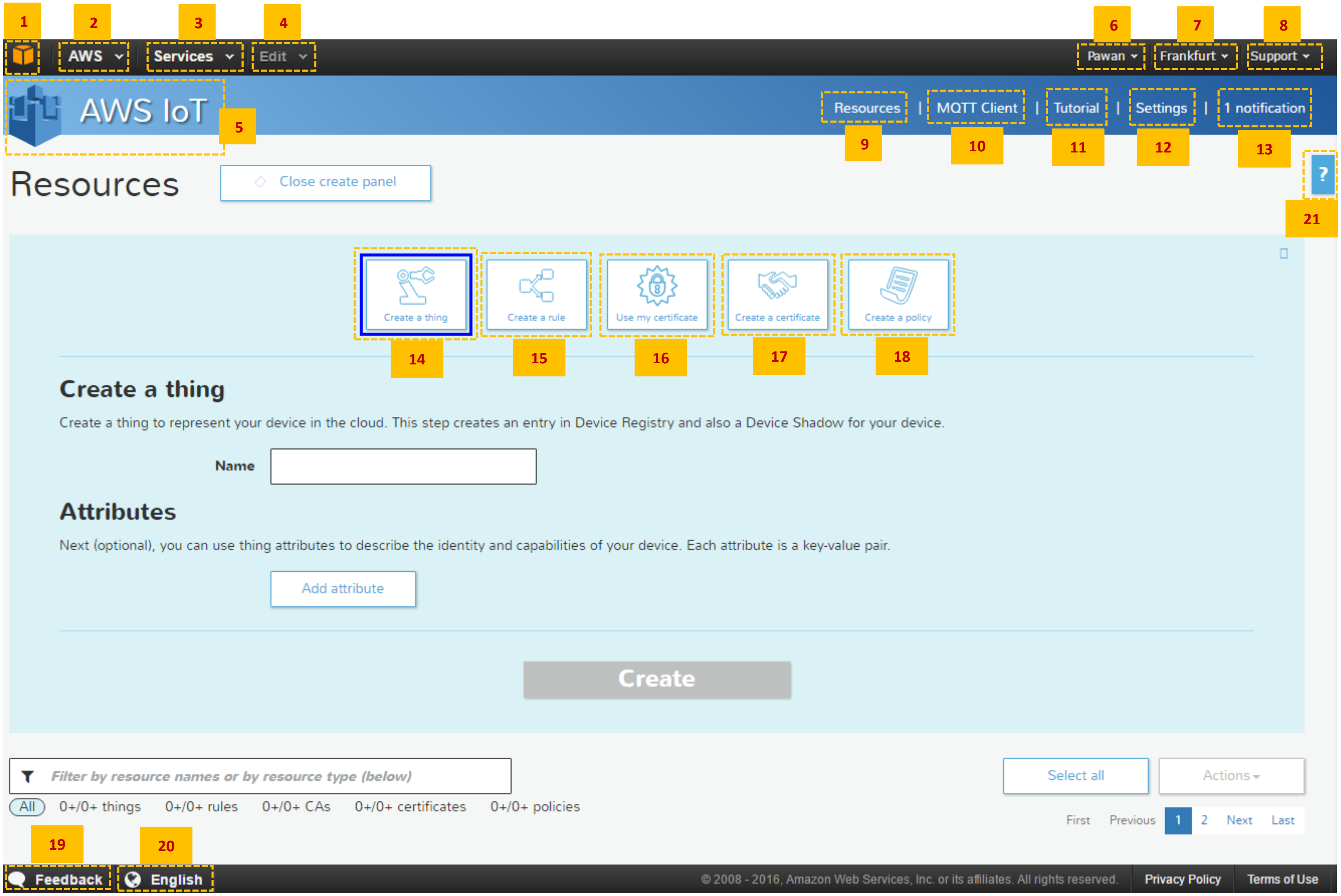
Microsoft Azure console (Portal)



	Microsoft Azure console (Portal): Description
1	Open dashboard panel to create new dashboard, edit existing dashboard, share, clone, delete dashboard.
2	List of existing dashboard(s).
3	Hide side menu.
4	Already opened service.
5	Search resources.
6	Notifications.
7	Customize dashboard. Add/Delete link to resources.
8	Portal settings: select a theme, enable/disable animations, command labels and task notifications, language preference, regional format etc.
9	Send a feedback.
10	Help + support, new support request, manage support requests, etc.
11	User profile information: logout, change password, permissions, submit an idea, view bills etc.
12	Add new resource(s).
13	Shortcut for different services (self-explanatory).
14	Browse from all the existing service.

Table 16: Microsoft Azure console (Portal) description

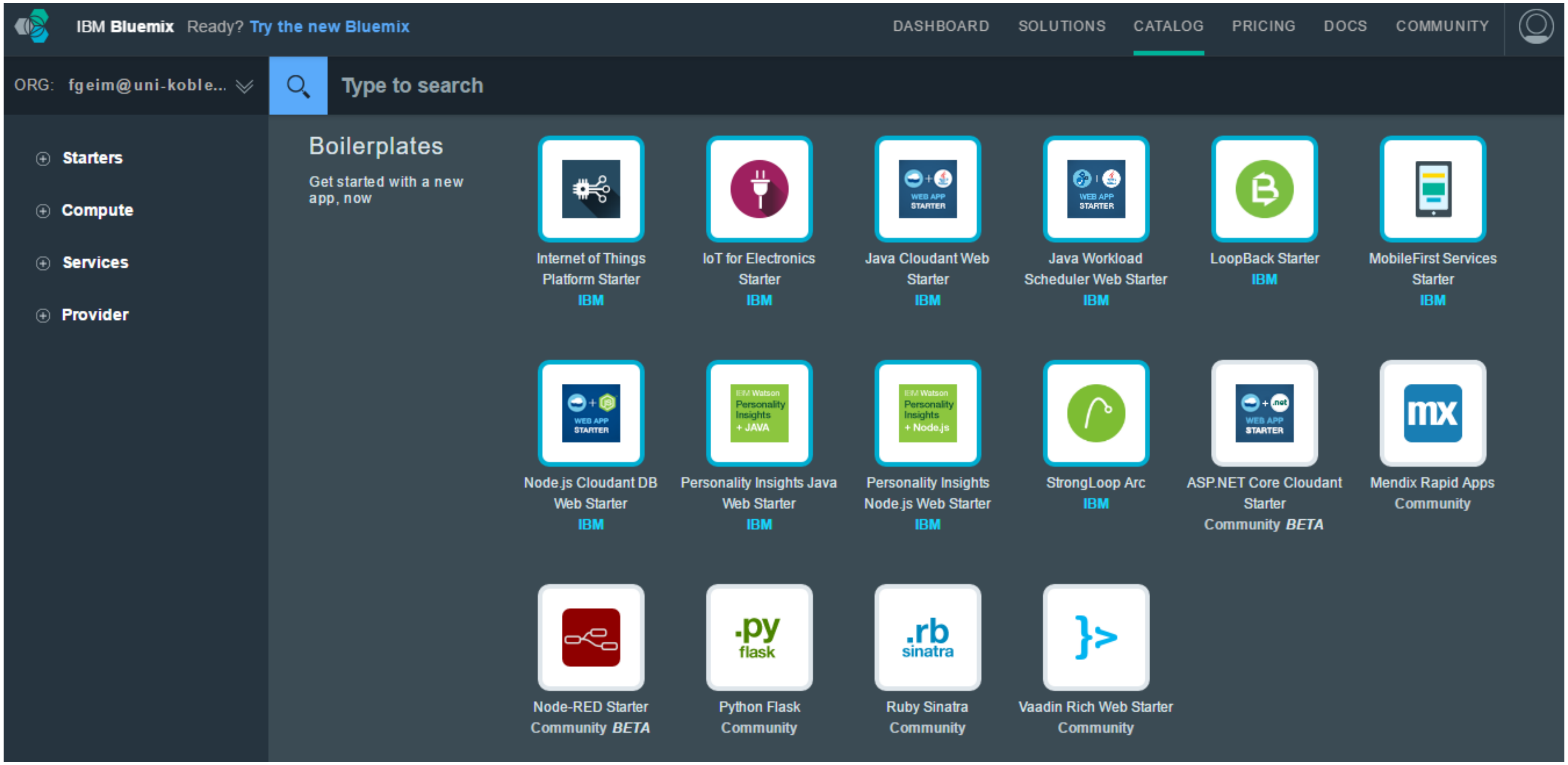
AWS IoT Console



	AWS IoT console: Description
1	Go to console home.
2	Create a resource group, or select from existing resource group(s). A resource group (Amazon) creates a custom console that manages the resources (i.e. services) we want to work with.
3	A list of all services that AWS offers.
4	Create one click shortcut of the services on the console.
5	Go to dashboard home.
6	User login information. It has the shortcuts to manage account, billing and cost, security credentials, etc.
7	Select a region from the list of available regions.
8	Support centre, forums, documentation, training, other resources.
9	Create IoT resource(s).
10	Set the MQTT client actions (need to be connected to the device gateway first).
11	Start interactive tutorials.
12	CloudWatch logs setting.
13	Notifications contains warnings or error messages during the activity on the console.
14	Create a “thing” by giving a unique name to it. This creates a device shadow for the device.
15	Create a rule for the incoming message from device.
16	Register CA certificate, or upload existing device certificate.
17	Create a certificate to authenticate device with AWS IoT.
18	Create a policy to define a set of authorized actions.
19	Send a feedback for AWS IoT console.
20	Change language preference.
21	Custom endpoint information.

Table 17: AWS IoT console description

IBM Bluemix Catalogue



Bibliography

Al-Fuqaha, A., Guizani, M., & Mohammadi, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials, 17*(4), 2347-2376.

Amazon. (2015, October). *New Services and Features Announced.* Retrieved from Amzon Web Services: https://aws.amazon.com/new/reinvent/

Amazon. (2016). AWT IoT API Reference. Amazon Web Services Inc. Retrieved from https://aws.amazon.com/documentation/iot/

Amazon. (n.d.-a). *Types of Cloud Computing*. Retrieved June 29, 2016, from Amazon AWS: https://aws.amazon.com/types-of-cloud-computing/

Amazon. (n.d.-b). *About AWS.* Retrieved June 19, 2016, from Amazon AWS: https://aws.amazon.com/about-aws/

Amazon. (n.d.-c). *AWS Partner Network.* Retrieved June 19, 2016, from Amazon AWS: https://aws.amazon.com/partners/

Amazon. (n.d.-d). *AWS IoT.* Retrieved June 19, 2016, from Amazon AWS: https://aws.amazon.com/iot/

Amazon. (n.d.-e). *Discussion Forum.* Retrieved from Amazon AWS: https://forums.aws.amazon.com

Amazon. (n.d.-f). *Logging AWS IoT API Calls with AWS CloudTrail.* Retrieved from Amazon AWS: http://docs.aws.amazon.com/iot/latest/developerguide/logging-using-cloudtrail.html

Andersson, P., & Mattson, L.-G. (2015). Service innovations enabled by the "internet of things". *IMP Journal, 9*(1), 85-106.

Ashton, K. (2009, June 22). That 'Internet of Things' Thing. *RFID Journal*, 1. Retrieved from http://www.rfidjournal.com/articles/view?4986

Basak, A. (2015, October 14). *Exploring the Microsoft Azure IoT Hub.* Retrieved from DZone/IoT Zone: https://dzone.com/articles/exploring-the-microsoft-azure-iot-hub

Beguelin, D. (2015, November). *Turn your smartphone into an IoT device.* Retrieved from IBM developerWorks: https://www.ibm.com/developerworks/library/iot-mobile-phone-iot-device-bluemix-apps-trs/

Betts, D. (2016, April 29). *Azure and Internet of Things.* Retrieved from Microsoft Azure: https://azure.microsoft.com/en-us/documentation/articles/iot-hub-what-is-azure-iot/

Bryant, A. W. (2015, October). *Living Roads.* New York: IBM Corporation. Retrieved from IBM: http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=AB&infotype=PM&htmlfid=YTC04027WWEN&attachment=YTC04027WWEN.PDF

Bullotta, R. (2011, September 7). *IoT Interview Series: 8 questions with Rick Bullotta of Thingworx.* Retrieved from Postscapes: http://postscapes.com/iot-voices/interviews/iot-interview-series-8-questions-rick-bullotta-thingworx/

Caldwell, J. (2015). IBM Internet of Things Point. INTERNATIONAL BUSINESS MACHINES CORPORATION. Retrieved from http://toronto.ieee.ca/files/2016/02/TOR-IEEE-IBM-IoT-Jan-28-2016-Final.pdf

Chen, Q., Hu, J., & Zhang, L. (2012). *A Study on Mobile Phone Security Industrial Ecology.* (Y. Wang, & X. Zhang, Eds.) Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-642-32427-7_57

Craig-Wood, K. (2010, February 24). *Definition of Cloud Computing, incorporating NIST and G-Cloud views.* Retrieved from katescomment: https://www.katescomment.com/definition-of-cloud-computing-nist-g-cloud/

Deloitte. (2014). *Industry 4.0 Challenges and solutions for the digital transformation and use of exponential.* Retrieved from http://www2.deloitte.com/content/dam/Deloitte/ch/Documents/manufacturing/ch-en-manufacturing-industry-4-0-24102014.pdf

Elastic. (n.d.). *Explore and Visualize your data.* Retrieved from Kibana: https://www.elastic.co/products/kibana

Elkstein, M. (2008). *Learn REST: A Tutorial.* Retrieved July 2016, 2016, from REST Blogs: http://rest.elkstein.org/

Eric Schmidt, R. K. (2015, August 12). *Announcing General Availability of Google Cloud Dataflow and Cloud Pub/Sub.* Retrieved from Google Cloud Platform Blogs: https://cloudplatform.googleblog.com/2015/08/Announcing-General-Availability-of-Google-Cloud-Dataflow-and-Cloud-Pub-Sub.html

Ericsson. (2016, June). *Ericsson Mobility Report.* Retrieved from Ericsson: https://www.ericsson.com/res/docs/2016/ericsson-mobility-report-2016.pdf

Evans, D. (2011, April). *The Internet of Things - How the Next Evolution of the Internet Is Changing Everything.* Retrieved from Cisco: http://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

Federal Ministry of Food and Agriculture. (2014). *Understanding Farming. Facts and figures about German farming.* Berlin: Federal Ministry of Food and Agriculture. Retrieved from http://www.bmel.de/SharedDocs/Downloads/EN/Publications/UnderstandingFarming.pdf?__blob=publicationFile

Fielding, R., Irvine, U., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., . . . Berners-Lee, T. (1999, June). Hypertext Transfer Protocol - HTTP/1.1.

Fowler, M. (2004). *UML Distilled: A Brief Guide to the Standard Object Modelling Language.* Pearson Education. Inc.

Gartner. (2013, November 11). *Gartner Says Personal Worlds and the Internet of Everything Are Colliding to Create New Markets.* Retrieved from Gartner Newsroom: http://www.gartner.com/newsroom/id/2621015

GEA. (2015). *Product catalogue – Barn technology for dairy farms. Practical system solutions for stalls, manure engineering and feeding.* Retrieved from GEA Farm Technologies: http://www.gea.com/global/en/binaries/DairyFarming_BarnEquipment_Catalog_0715_tcm11-27813.pdf

Google. (n.d.-a). *Products and Services.* Retrieved June 24, 2016, from Google Cloud Platform: https://cloud.google.com/products/

Google. (n.d.-b). *Brillo.* Retrieved June 23, 2016, from Google Developers: https://developers.google.com/brillo/

Google. (n.d.-c). *Overview of Internet of Things.* Retrieved June 29, 2016, from Google Cloud: https://cloud.google.com/solutions/iot-overview

Google. (n.d.-d). *Regions and Zones.* Retrieved June 23, 2016, from Google Cloud Platform: https://cloud.google.com/compute/docs/regions-zones/regions-zones

Google. (n.d.-e). *Join a Google Cloud Platform community.* Retrieved June 24, 2016, from Google Cloud Support: https://support.google.com/cloud/answer/3466163

Google. (n.d.-f). *Google Cloud SDK*. Retrieved July 6, 2016, from Google Cloud: https://cloud.google.com/sdk/

Google. (n.d.-g). *Cloud Source Repositories.* Retrieved June 24, 2016, from Google Cloud: https://cloud.google.com/source-repositories/

Govloop. (2015). *The Internet of Things What the IoT means for the Public Sector.* Washington, DC: Govloop. Retrieved from http://www.isaca.org/Groups/Professional-English/cybersecurity/GroupDocuments/IoT%20in%20the%20Public%20Sector.pdf

Hevner, A. R. (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems, 19*(2), 87-92.

Hevner, A. R., & Chatterjee, S. (2010). *Design Research in Information Systems: Theory and Practice.* Springer Science and Business Media.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly, 28*(1), 75-105.

HiveMQ. (n.d.). *MQTT 101 - How to Get Started with the lightweight IoT protocol.* Retrieved July 16, 2016, from HiveMQ Blog: http://www.hivemq.com/blog/how-to-get-started-with-mqtt

IBM. (2014, June 20). *IBM's Bluemix now generally available - significant momentum since open beta.* Retrieved from IBM Press Release: https://www-03.ibm.com/press/us/en/pressrelease/44239.wss

IBM. (2016-a). *Reference material IBM's IoT Platform service.* Retrieved June 29, 2016, from IBM IoT Foundation: http://iotf.readthedocs.io/en/latest/

IBM. (2016-b, September 9). *Bluemix Overview*. Retrieved July 12, 2016, from Bluemix Docs: https://console.ng.bluemix.net/docs/overview/index.html

IBM. (2016-c, July 28). *DeveloperWorks Recipes - Visualizing your data.* Retrieved from IBM DeveloperWorks: https://developer.ibm.com/recipes/tutorials/visualizing-your-data/

IBM. (2016-d). *Understanding the impact and value of enterprise asset management.* New York: IBM Corporation. Retrieved from http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=TIB14016USEN

IBM. (n.d.-a). *Watson Internet of Things Partners.* Retrieved June 27, 2016, from IBM: http://www.ibm.com/internet-of-things/ecosystem/partners/

IBM. (n.d.-b). *IBM Watson IoT Platform*. Retrieved June 29, 2016, from IBM Cloud: https://internetofthings.ibmcloud.com/#/

IBM. (n.d.-c). *Maximo asset management.* Retrieved August 28, 2016, from IBM Software: http://www-03.ibm.com/software/products/en/maximoassetmanagement

IEEE. (2015, January). *IEEE Standards Association (IEEE-SA) Internet of Things (IoT) Ecosystem study.* Retrieved from IEEE standard association: http://standards.ieee.org/innovate/iot/study.html

IEEE. (2016). *P2413 - Standard for an Architectural Framework for the Internet of Things (IoT).* Retrieved August 4, 2016, from IEEE Standards Association: https://standards.ieee.org/develop/project/2413.html

IETF. (2014, June). *The Constrained Application Protocol (CoAP).* Retrieved from IETF TOOLS: https://tools.ietf.org/html/rfc7252

Iivari, J. (2015). Distinguishing and contrasting two strategies for design science research. *European Journal of Information Systems, 24*(1), 107-115.

Impe, P. V. (2014, November 6). *You are invited – The business value of the 'Internet of things'.* Retrieved from The Data Science Community: https://datasciencebe.com/2014/11/06/you-are-invited-the-business-value-of-the-internet-of-things/

Inquirer. (2015). *INQUIRER Tech Hero Awards.* Retrieved from The Inquirer: http://www.theinquirer.net/inquirer/feature/2406764/inquirer-tech-hero-awards

IoT-A. (2013). *Internet of Things – Architecture.* Retrieved from IoT-A : http://www.iot-a.eu/public/public-documents/

Kaazing Corporation. (n.d.). *HTML5 WebSocket: A Quantum Leap in Scalability for the Web*. Retrieved July 16, 2016, from WebSocket: http://www.websocket.org/quantum.html

Karasiewicz, C. (2013, September). *Why HTTP is not enough for the Internet of Things.* Retrieved from IBM developerWorks: https://www.ibm.com/developerworks/community/blogs/mobileblog/entry/why_http_is_not_enough_for_the_internet_of_things?lang=en

Kidman, A. (2014, December 10). *Lifehacker Awards 2014: Best Cloud Computing Provider.* Retrieved from Lifehacker: http://www.lifehacker.com.au/2014/12/lifehacker-awards-2014-best-cloud-computing-provider/

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Design Support System, 15*, 251-266.

Marechaux, J.-L. (2015, July 7). *Top 10 reasons to use Bluemix and DevOps Services.* Retrieved from IBM DeveloperWorks: http://www.ibm.com/developerworks/library/d-10-reasons-to-use-bluemix-devops-trs/index.html

Mazhelis, O., Luoma, E., & Warma, H. (2012). Defining an Internet-of-Things Ecosystem. In *Internet of Things, Smart Spaces, and Next Generation Networking: 12th International Conference, NEW2AN 2012, and 5th Conference, ruSMART 2012, St. Petersburg, Russia, August 27-29, 2012. Proceedings* (pp. 1-14). Berlin, Heidelberg: Springer Berlin Heidelberg.

Mell, P., & Grance, T. (2011, September). *The NIST Definition of Cloud Computing.* Retrieved from NIST: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

merriam-webster. (n.d.). *Dictionary.* Retrieved from Merriam-Webster's Learner's Dictionary: http://www.merriam-webster.com/dictionary/animal%20husbandry

Microsoft. (2010, February 1). *Windows Azure Platform Now Generally Available in 21 Countries.* Retrieved from Microsoft Azure Blogs: https://azure.microsoft.com/en-us/blog/windows-azure-platform-now-generally-available-in-21-countries/

Microsoft. (2016-a). *Internet of Things: Engage with an Azure IoT partner*. Retrieved June 29, 2016, from Azure IoT Partners: http://azureiotpartners.azurewebsites.net/#countryFilter=global&industryFilter=general---applicable-to-all&partnerCategoryFilter=devices

Microsoft. (2016-b, April 27). *Azure IoT Suite goes international.* Retrieved from Microsoft Azure: https://azure.microsoft.com/en-us/updates/azure-iot-suite-goes-international/

Numoto, T. (2015, March 16). *Microsoft announces Azure IoT Suite.* Retrieved from Microsoft Blogs: https://blogs.microsoft.com/iot/2015/03/16/microsoft-announces-azure-iot-suite/

OASIS. (2012, October 12). *OASIS Advanced Message Queuing Protocol (AMQP) Version 1.0.* Retrieved from OASIS Standards: http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-complete-v1.0-os.pdf

Orlikowski, W. J., & Scott, S. V. (2008). Sociomateriality: Challenging the separation of technology, work and organization. *Academy of Management Annals, 2*(1), 433-474.

Postscapes. (2014-a). *Internet of Things Awards.* Retrieved from Postscapes: http://postscapes.com/internet-of-things-award/2014/iot-platform-tools.html

Postscapes. (2014-b). *Internet of Things Awards.* Retrieved from Postscapes: http://postscapes.com/internet-of-things-award/2014/winners.html

Postscapes. (2016). *Internet of Things Awards.* Retrieved from Postscapes IoT Awards: http://iotawards.postscapes.com/2015-16/best-iot-cloud-platform

postscapes. (n.d.). *Smart City Applications*. Retrieved August 21, 2016, from Postscapes - Internet of Things awards: http://postscapes.com/internet-of-things-award/smart-city-application/

PTC. (2015, January 14). *PTC's ThingWorx and Axeda Internet of Things (IoT) Enabling Technologies Win Multiple Excellence Awards at CES 2015.* Retrieved from PTC News: http://www.ptc.com/news/2015/thingworx-axeda-win-awards-ces-2015

Reuters. (n.d.-a). *Profile: International Business Machines Corp (IBM).* Retrieved June 27, 2016, from Reuters: http://www.reuters.com/finance/stocks/companyProfile?symbol=IBM

Reuters. (n.d.-b). *Amazon.com Inc (AMZN.O).* Retrieved June 19, 2016, from Reuters Finance: http://www.reuters.com/finance/stocks/companyProfile?symbol=AMZN.O

Robot shop. (n.d.). *Linear Actuators.* Retrieved August 20, 2016, from Robot shop: http://robotshop.com/en/actuators.html

Rockey, J. W. (1965). *Loose-housing system for Dairy Cattle.* US Department of Agriculture.

Rose, K., Eldridge, S., & Chapin, L. (2015). *The internet of things: An overview.* The Internet Society (ISOC).

Scientific Advisory Board on Agricultural Policy. (2015). *Pathways to a socially accepted livestock husbandry in Germany. Executive Summary and Synthesis Report.* Berlin. Retrieved from http://www.bmel.de/EN/Ministry/Scientific-Advisory-Boards/_Texte/AgriculturalPolicyPublications.html

Seeed. (2015). *Grove Indoor Environment Kit for Edison.* Retrieved August 25, 2016, from Seeed Wiki: http://wiki.seeedstudio.com/wiki/Grove_Indoor_Environment_Kit_for_Edison

Sheng, Z., Yang, S., Yu, Y., Vasilakos, A. V., McCann, J. A., & Leung, K. K. (2013, December). A Survey on the IETF protocol suite for the Internet of Things: Standards, Challenges, and Oppertunities. *IEEE Wireless Communication*, 91-98. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6704479

Singh, N. (2016, February 8). *Azure IoT Hub general availability overview.* Retrieved from Microsoft Azure Blogs: https://azure.microsoft.com/en-us/blog/azure-iot-hub-ga-capability-overview/

Sundmaeker, H., Guillemin, P., Friess, P., & Woelfflé, S. (2010). *Vision and challenges for realising the Internet of Things.* Cluster of European Research Projects on the Internet of Things, European Commision. Retrieved from http://www.internet-of-things-research.eu/pdf/IoT_Clusterbook_March_2010.pdf

ThingWorx. (n.d.-a). *ThingWorx Marketplace.* Retrieved June 27, 2016, from ThingWorx Partners: http://marketplace.thingworx.com/partners

ThingWorx. (n.d.-b). *Getting Started with ThingWorx 7.0.* Retrieved June 27, 2016, from ThingWorx: http://support.ptc.com/cs/help/thingworx_hc/thingworx_7.0_hc/

ThingWorx. (n.d.-c). *ThingWorx Foundation Core Concepts.* Retrieved June 27, 2016, from ThingWorx Developer Dashboard: http://developer.thingworx.com/core-concepts/

Trebilcox-Ruiz, P. (2016, February 26). *Introducing Brillo and Weave.* Retrieved from envatotuts+: http://code.tutsplus.com/tutorials/introducing-brillo-and-weave--cms-25859

Usländer, T. (2014, March). The Trend towards the Internet of Things: What does it help in Disaster and Risk Management? *5th IDRC Davos . 3*, pp. 726-729. GRF Davos Planet@Risk. Retrieved from https://planet-risk.org/index.php/pr/article/view/189

Vaishnavi, V. K., & Kuechler, W. (2008). *Design Science Research Methods and Patterns.* Auerbach.

Ventana Research. (2015). *Ventana Research Announces the 2015 Technology Innovation Awards.* Retrieved from Ventana Research : http://www.ventanaresearch.com/2015TechnologyInnovationAwardsPR/

Williams, A. (2015, August 20). *DevBoard Watch: BeagleBone Green showcases Grove sensors.* Retrieved from Electronics Weekly: http://www.electronicsweekly.com/blogs/gadget-master/beagleboard/devboard-watchbeaglebone-green-showcases-grove-sensors-2015-08/

yadav, R. (2016, May 3). *How the weather company delivers the worl's weather.* Retrieved from developerWorks: https://developer.ibm.com/dwblog/2016/how-the-weather-company-delivers-the-worlds-weather/