Chapter 1
Cloud Computing and Internet of Things Integration: Architecture, Applications, Issues, and Challenges

Akash Malik and Hari Om

1.1 Introduction

Cloud computing and IoT are two distinct technologies having wide applications in human life. Their acquisition and use is extremely pervasive, making them future of internet. The IoT is the internetworking of physical devices, embedded with electronics, software, sensors, actuators, and network connectivity, that enable these objects to collect and exchange the data. IoT describes a system where the items in physical world, and sensors within or attached to these items, are connected to the Internet via wireless and wired Internet connections. These sensors can use different types of local area connections such as Radio-Frequency Identification (RFID), Near Field Communication (NFC), Wireless Fidelity (Wi-Fi), Bluetooth, and Zigbee. The sensors can also have wide area connectivity such as Global System for Mobile communication (GSM), General Packet Radio Service (GPRS), 3G, and Long Term Evolution (LTE) [1]. IoT mainly divided into three components that enable seamless ubiquitous computing: (1) hardware—made up of actuators, sensors, and embedded hardware for data exchange; (2) middleware—provides data storage and computing applications based on demand for data analytics; and (3) presentation—an easily understandable visualization and interpretation tools that can be accessed on different platforms and build for various applications [2]. Cloud computing is a type of on-demand Internet-based computing model consisting of autonomous, networked IT (hardware and/or software) resources. It may be considered as the delivery of power of computer as needed, information storage, applications, and other tools of information technology by cloud services platform via the internet with pay-per-use pricing [3]. The service providers offer

A. Malik (✉) • H. Om
Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, India
e-mail: akashmalik28@gmail.com; hariom4india@gmail.com

W. Rivera (ed.), *Sustainable Cloud and Energy Services*,
DOI 10.1007/978-3-319-62238-5_1

cloud services through the Internet as a set of easy-to-use, scalable, and economical services to interested clients on subscription basis.

The IoT may be specified by the small things of real world that are extensively distributed and have constrained storage capacity as well as processing power. Due to these constraints, it has issues related to reliability, performance, privacy, and security. Additionally, there exists large amount of heterogeneity in data as well as devices and IOT provides a platform to handle all of them. On the contrary, the cloud computing, a mature technology, provides almost unlimited capabilities in terms of storage as well as processing power, and has helped addressing most of the IoT issues. So, a paradigm consisting of cloud and IoT, two interdependent technologies, together is expected to disorder both current and future Internet [4]. The term given for Cloud and IoT integration is *CloudIoT* or *Cloud of Things* (CoTs). Cloud Computing and IoT integration explores the area of IoT capabilities as well as cloud capabilities. Furthermore, new capabilities like data mining, complex analysis, and real life processing in IoT as well as cloud service area will be explored.

1.1.1 Internet of Things

The terminology *Internet of Things* (IoT) was first introduced in 1999 by British technology pioneer Kevin Ashton to describe a network system where the objects in physical world can be connected to the Internet by sensors. Ashton further devised this term to highlight the capability of connecting the RFID tags to the Internet. In IoT, "things" refer to any object on the face of the Earth irrespective it is a communicating device or a non-communicating object. Today, the IoT has become a trendy term in which the Internet connectivity and computing capability are continued to various categories of objects, devices, sensors, and usual items. Despite universal acceptance of IoT, there is no single globally recognized definition for the term. There are various definitions of IoT [1] and are as follows:

> The term Internet of Things belongs to scenarios in which connectivity of the network and computing capability strikes to objects, sensors, and usual items not personal computers, granting these devices to create, interchange, and utilize data with minimum human involvement.

> The Oxford Dictionaries definition:

> Internet of Things (noun): The interconnection via the Internet of computing devices embedded in everyday objects, enabling them to send and retrieve data.

> The RFID group defines Internet of Things as:

> The worldwide network of interconnected objects uniquely addressable based on standard communication protocols.

1.1.1.1 IoT Devices, Platform, and Services

We can divide IoT devices into two comprehensive classes: wearable and microcontroller/microprocessor driven embedded IoT devices. Some examples according to categories of hardware devices in IoT are given below.

Wearable Devices Google Glass, Samsung Gear 2 neo, Pebble Watch, Misfit Shine, Android wear, etc.

Embedded System-Based Devices Arundio, Intel Galileo, Raspeberry-Pi, Beglebone, etc.

Accessories Relays, Displays, Switches, Actuators, Sensors, etc.

Smart Devices An object is a smart object that can define its own possible interactions. Any object which has a state as well as has certain information joined with that state which can also determine type of connectivity, time span of connectivity, and connectivity protocol are called smart objects. Smart objects in IoT are Global Positioning System (GPS), Global System for Mobile communications GSM, General Packet Radio Service (GPRS), Radio Frequency Identification (RFID), Near Field Communication (NFC), etc.

Platform and Services Some platform and services of IoT are RIOT, Carriots, Lithouse, Sensinode, IFTTT, Arrayent, Alljoyn, ioBridge, Tizen, SalesForce, Axeda, OpenIoT, etc.

1.1.1.2 Layered Architecture of IoT

On the bases of layered architecture, the IoT is divided into four layers: perception/link layer, network/transport layer, middleware layer, and application/service layer, shown in Fig. 1.1 [5]. These layers are described below.

Perception Layer/Link Layer

The Perception Layer/Link Layer is the bottom layer where sensors, RFID, etc., devices are launched into the IoT network to sense the happening in environment and report to sink nodes via network layer. This layer further divided into two layers: edge technology and access gateway layers, which are discussed below.

(a) *Edge technology layer*: This is a hardware layer comprising embedded systems, sensor networks, RFID tags, and many other different sensors. This layer performs many other respective functions such as gathering information from a system or an environment, processing this information, and assisting communication.
(b) *Access gateway layer*: This layer deals with handling of data generated by IoT devices or things. It is primarily accountable for publishing, and subscribing the services provided by the IoT devices, message routing, and hovelling communication between platforms.

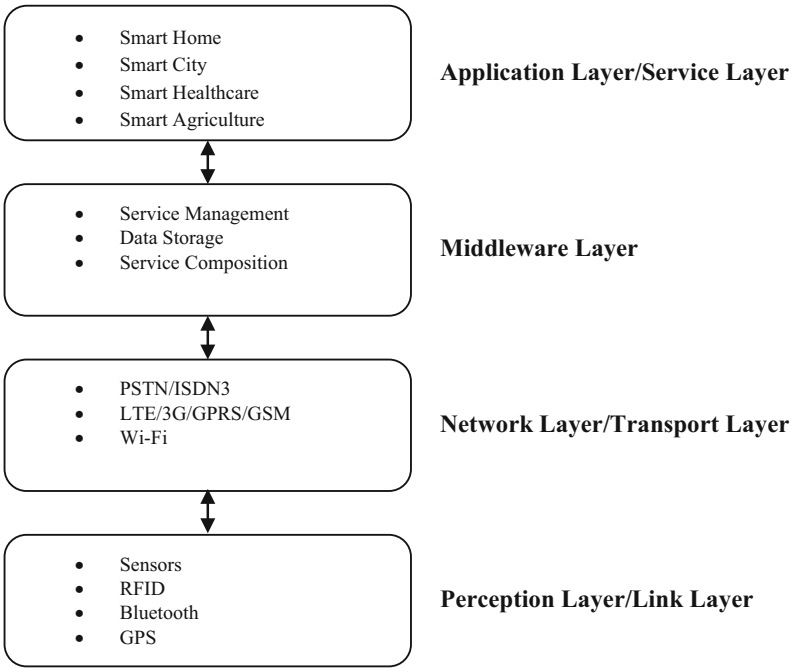

Fig. 1.1 Layered architecture of IoT

Network Layer

This layer may be considered a group of internetworking methods, protocols, and specifications in the Internet protocol suite, which can be utilized to transport datagrams (packets) from a sending host across the network boundaries to a receiving host. The sender and receivers are mentioned by the network addresses (IP addresses) by the Internet Protocol (IP).

Middleware Layer

This layer performs collecting and filtering the data received from the hardware devices, data discovery, and access control to the devices for applications. This layer can be located anywhere like in sensor nodes, sink nodes, high-level application terminal, etc. The main goal of middleware layer is to provide abstraction of functionalities and communication capabilities of the devices in IoT deployment for achieving a ubiquitous integration with other technologies like cloud services [6].

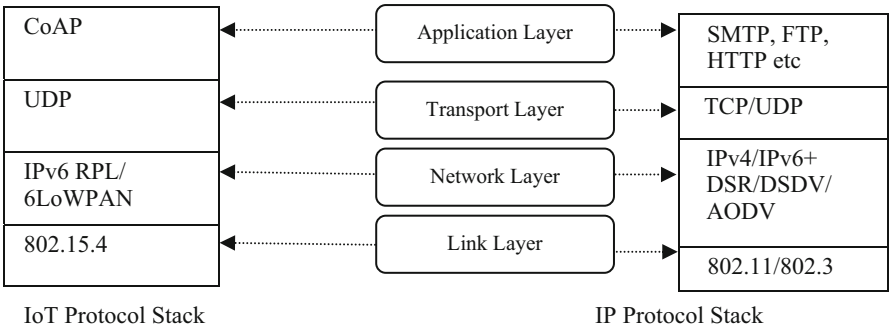

Fig. 1.2 IoT network and traditional IP network

Application Layer

In IoT-based systems, application layer is used for delivering different application services provided over middleware layer to different users and applications. The application services can be used in many industries like healthcare, logistics, retail, etc.

1.1.1.3 Comparison of IoT Network and Traditional IP Network

The IoT devices are resource constrained, i.e., they have limited battery, limited processing power, low transmission range, etc., and also the network links are lossy. The module of traditional IP protocol is heavyweight and hence is not suitable for IoT system architecture. The lightweight protocol corresponding to the traditional IP protocol for resource-constrained devices in IoT has been developed [5, 7]. Figure 1.2 shows the layers in IoT protocol stack vs. IP protocol stack.

(a) *Constrained Application Protocol (CoAP)*: Internet Engineering Task Force (IETF) has developed the Constrained Application Protocol (CoAP), an application protocol for IoT devices, to connect the IoT devices to Internet. It is a replacement of the HTTP protocol for application layer of IoT, considering optimized length of datagram and satisfying the representational state transfer (REST) protocol to support Uniform Resource Identifier (URL). It provides a reliable communication based on user datagram protocol (UDP).

(b) *User Datagram Protocol (UDP)*: The UDP is a lightweight connectionless protocol, suitable for resource-constrained IoT devices, that provides checksum for data integrity and port number to communicate with other nodes. Since the IoT applications are time sensitive, dropping of packets is preferred rather than waiting for delayed packet. The UDP protocol satisfies this requirement and hence has been adopted for transport layer protocol.

(c) *Link Layer Protocol*: The IoT implements IEEE 802.15.4 standard protocol for sensor devices for medium access control (MAC) layer. The frame formats of a traditional network at link layer are not suitable for resource-constrained devices in IoT due to their overhead. The IEEE 802.15.4 defines frame format, header, and communication algorithms for IoT devices that uses channel hopping and time synchronization. Its other features are slotted frame structures, synchronization, channel hopping, network formation, scheduling. Some other MAC layer protocols for IoT applications are IEEE 802.11 AH, wireless HART, Bluetooth low energy, Zigbee smart energy, DASH7, Home plug, G.9959, LTE-A, LoRaWAN, Weightless, etc.

(d) *Network Layer Protocol*: IoT uses IPv6 routing to overcome the address space problem in IoT, and lossy network (RPL) and 6LoWPAN network at network layer for resource-constrained sensor nodes.

(e) *RPL (Routing Protocol for low power and Lossy network)*: The RPL, primarily designed to meet the specific requirement of IP IoT, is a novel standard routing protocol that can perform one-to-one, one-to-many, and many-to-one communication. It supports both unidirectional and bidirectional communication between the root, also called sink node, and constrained nodes for creating a destination oriented directed acyclic graph (DODAG) with the root node. The root node is directly connected to the Internet using IPv6 Boarder Router (6BR) using a three-way handshake as given below.

- *DODAG information object (DIO) message*: The root node broadcasts the DIO message for the formation of topology.
- *DODAG advertisement object (DAO) message*: Other nodes select their parents after receiving the DIO messages and reply a DAO message to the parent asking the permission to join it.
- *DODAG acknowledge message (DAO ACK)*: Parent node permits by sending DAO ACK message based on the rank value calculation for each node using the rank (of possible parent) and other parameters like node energy and node distance. If a new node wants to join a parent, it sends a DODAG Info solicitation (DIS) message to find if any DODAG exists.

(f) *IPv6 over Low Power Wireless Personal Area Network (6LoWPAN)*: The 6LoWPAN integrates Wireless Sensor Networks (WSNs) and IP-based infrastructure such as IEEE 802.15.4 networks and performs context aware header compression as given below:

- IP Header Compression (IPHC) to compress IPv6 header.
- Next Header Compression (NHC) to compress the user datagram protocol (UDP) header and IPv6 extension header.

The 6LoWPAN standard redefines fragmentation and reassembly of packets because the payload size of link layer in 6LoWPAN network is very limited and the security protocol for large application data size makes the IEEE 802.15.4 frame size larger than the maximum transmission unit (MTU) size (127 bytes). The 6LoWPAN fragmentation scheme provides reassembly tag and an offset to every fragment and provides additional fragmentation in case the data size exceeds MTU size.

1.1.2 Cloud Computing

Cloud computing, also known as on-demand computing, is an internet-based computing, where a shared pool of resources and information is provided to computers and other devices on demand [3].

The cloud computing has following characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured services.

- *On-demand self-service*: A consumer can unilaterally provision computing capabilities such as server time and network storage as per requirement automatically without requiring human interaction.
- *Broad network access*: Various capabilities are available over the network that can be accessed through standard mechanisms to promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).
- *Resource pooling*: The computing resources such as storage, processing, memory, and network bandwidth are pooled to serve multiple consumers using a multi-tenant model in which the physical and virtual resources of different types are dynamically assigned and reassigned as per demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter).
- *Rapid elasticity*: It refers to the capabilities that can allocate and release the resources as per demand automatically at any time.
- *Measured service*: Cloud systems automatically control and optimize the resource usage by leveraging a metering capability at some level of abstraction, e.g., storage, processing, bandwidth, and active user accounts. Resource usage can be monitored, controlled, and reported in transparent manner for both the provider and consumer.

1.1.2.1 Services of Cloud Computing

There are three main services provided by cloud computing as given below [3].

(a) *Platform as a Service (PaaS)*: The PaaS is offered as a development environment to an application developer through a toolkit. With Platform as a Service, one (user) deploys self-created or acquired applications onto the cloud infrastructure using the tools and the programming language provided by the provider. The user does not bother about the underlying cloud infrastructure such as network, operating systems, or storage, but he can access and use the deployed applications and the application hosting environment configurations. Google provides this type of service where users create and host the web applications using Google's development.

(b) *Software as a Service (SaaS)*: The SaaS model helps using the applications running on a cloud infrastructure, which can be accessed from different user devices through a thin client interface such as web browser and a program interface. The web-based email and Salesforce, an online sales management, are examples of Software as a Service.
(c) *Infrastructure as a Service (IaaS)*: An IaaS provider provides virtual or physical systems/machines and other resources as per the Internet Engineering Task Force (IETF). In IaaS, one is able to access processing power, storage, and other computing resources. The user does not bother about the underlying cloud infrastructure, but has accessibility over the operating systems, storage, deployed applications, and limited accessibility over certain networking components, such as host firewalls. Amazon provides this type of service where users can rent virtual servers on which they run their own applications.

1.1.2.2 Deployment Models

There are mainly four types of deployment models in cloud environment.

(a) *Private Cloud*: It is the cloud infrastructure provisioned for the use of a single organization with multiple consumers, and hosted either internally or externally, for example, individual business unit in a large corporation. The private cloud may be handled including its ownership and manageability by the organization or a third party, or some of them, and it can be on or off premises.
(b) *Public Cloud*: It is the cloud that provides the services (which may be free) using a network for public use on the premise of the cloud provider, for example, a university.
(c) *Community Cloud*: It is the cloud that is exclusively used by a specific community from organizations having common concerns such as security requirements and policy. It may exist on or off premises and managed by one or more organizations, a third party, or some combination of them.
(d) *Hybrid Cloud*: It is a composition of two or more clouds (private, community, or public), each maintaining its unique entity. However, they together offer collectively the benefits of multiple deployment models.

The remaining chapter is arranged as follows. In Sect. 1.2, the works related to Cloud and IoT integration are discussed. In Sect. 1.3 Cloud and IoT integration architecture components and architecture based on CoAP and 6LowPan are given. In Sect. 1.4 various applications of integration discussed. Section 1.5 points out challenges and open issues. Section 1.6 provides conclusions and future work.

1.2 Related Works

The cloud and IoT have independently seen a rapid evolution, which may be considered complementary to each other and researchers are working on their integration [6, 8–11]. Their integration has storage processing and networking capabilities that mainly come under IoT due to its characteristics. The devices related to IoT contain a lightweight and compatible mechanism for communication with the cloud systems deployed through IoT middleware. Since IoT has devices, technologies, and protocols of different types, its functionalities like scalability, interoperability, reliability, efficiency, etc., are quite challenging to get. Its integration with cloud addresses these types of issues [12, 13] and also provides many other features like ease-of-access, ease-of-use, etc. [12]. Since IoT has limitations in terms of storage, processing, and communication, the cloud computing can help in overcoming these constraints. The cloud can provide an effective solution for IoT service management and implementing applications and services that exploit the things or data produced by them [9]. The cloud can also be benefitted from IoT by increasing its scope to manage real-world things. It will certainly affect the application development in future in which gathering, processing, and transmission of information will cause further issues [10].

The adoption of cloud and IoT integration, which is also termed as CloudIoT paradigm, helps enabling new scenarios for smart services and applications based on the cloud through the things: Sensing as a Service (SaaS) that can provide pervasive access to the user data [14], Sensing and Actuation as a Service (SAaaS) that can provide automatic control logics in a cloud [14], Sensor Event as a Service (SEaaS) that can enable to dispatch messaging services triggered by the sensor events [14], Sensor as a Service (SenaaS) that can provide pervasive management of remote sensors [15], DataBase as a Service (DBaaS) that can provide pervasive database management [15], Data as a Service (DaaS) that can provide pervasive access to any kind of data [15], Ethernet as a Service (EaaS) that can provide pervasive layer-2 connectivity to remote devices [15], Identity and Policy Management as a Service (IPMaaS) than can provide pervasive access to policy and identity management functionalities [15], Video Surveillance as a Service (VSaaS) that can provide pervasive access to recorded video and implementing complex analyses in the cloud [16]. These services have led to many applications such as healthcare, smart home and smart metering, video surveillance, automotive and smart mobility, smart logistics, and environmental monitoring [15]. Deployment of RFID applications in IoT networks is generally complex and costly as they require tedious deployment and management of large and heterogeneous distributed systems [17]. The RFID applications are suitable for large organizations only as small business organizations have limited resources. This issue can be addressed by cloud computing with integration of virtualization technologies and the architecture of web and its services [17]. There are various protocols for integration of cloud and IoT that take care of research oriented and open-source related projects and enterprise products in variety of disciplines. The Nimbits [18], ThingSpeak [19], Paraimpu [20], DeviceCloud

[21], SensorCloud [22], iDigi Device Cloud [23], Stack4Things [24] are some of such protocols. There are some Things development platforms that include Wiring [25], Sun SPOT [26], mbed [27], or Arduino [23].

The architecture of cloud and IoT integration using the combination of Hypertext Transfer Protocol (HTTP) and Message Queuing Telemetry Transport (MQTT) protocol servers has been discussed in [28]. A common integration architecture of cloud and IoT, which is called CloudThings, based on RESTFul web services, and protocols like CoAP, for communication between constrained resources things and constrained networks, and IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) protocol has been discussed in [29]. The extension of cloud, which is closer to things that produce and acts on the data in IoT, is termed as Fog Computing. Integration architecture of Fog, Cloud, and IoT has been discussed in [30]. There are several IoT applications such as e-health, Smart Grid and Power Systems, generating a large amount of data regularly. Therefore, a secure data analytics for cloud and IoT integration (cloud-Integrated Internet of Things) applications is needed. A reference architecture for privacy reservation in cloud-based IoT applications has been discussed in [31] and fully homomorphic encryption systems to achieve data security and privacy over cloud-based analytical phases on three applications: (1) abnormality detection and patient classification in electronic health records, (2) anomaly detection services for WSN-based IoT monitoring, (3) smart grid billing services, etc., have been discussed in [32].

1.3 IoT and Cloud Computing Integration Architecture

The cloud and IoT contain many characteristics which are complementary to each other as shown in Table 1.1. So, their integration can provide good solutions to real-world problems. The IoT can gain from virtually unlimited storage resources be it storage or computing, and pervasive reachability of Cloud. The cloud can benefit from IoT by increasing its scope to address the real-world issues by providing new services in real life scenarios [11].

Table 1.1 Characteristics of IoT and Cloud

Cloud	IoT	Characteristics
Provide techniques to manage	Generator	Big data
Virtually unlimited	Limited or none	Storage capacity
Virtually unlimited	Limited	Computational capabilities
Centralized	Widespread	Displacement
pervasive	Limited	Reachability
Mode of service providing	Point of convergence	Internet role

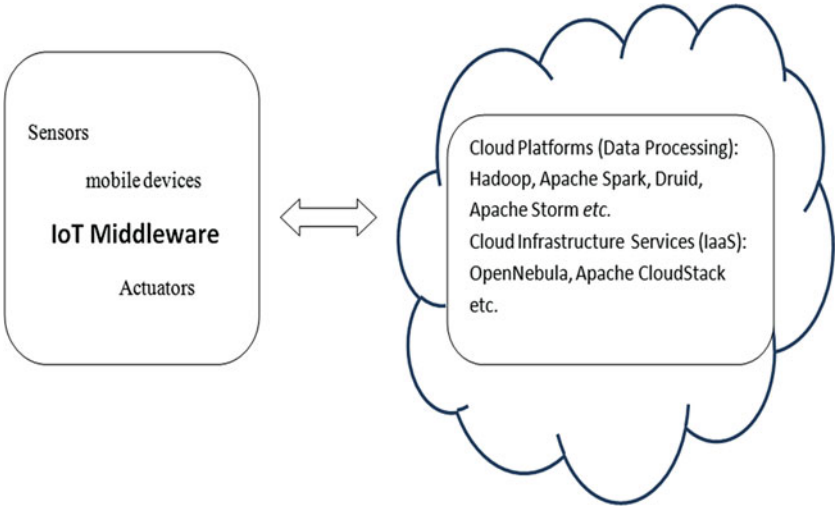

Fig. 1.3 Cloud and IoT integration paradigm

1.3.1 IoT and Cloud Integration Components

Integration components of cloud and IoT may be classified into three categories considering seamless integration that include cloud platforms and infrastructure, and IoT middleware. The cloud platforms address the IoT limitations and provide business opportunities and additional scalability. Cloud platforms are managed by the Cloud Infrastructures by deploying and monitoring. Middleware of the IoT provides a lightweight and interoperable procedure for exchanging data between the IoT devices and Cloud system deployed. Figure 1.3 shows the integration components [33].

1.3.1.1 Cloud Platforms

Since the IoT users are increasing every day, the data generated by them is also increasing, which has made the database management systems (DBMS) inappropriate. Therefore, a platform with high scalability, storage, and processing power is needed. We study various platforms for storing large amount of data as well as processing that data.

(a) *Hadoop*: Hadoop [34, 35] is a freely available software framework, which manages big data by distributed storage and processing using MapReduce programming model. It is divided into four modules: Hadoop Common, Hadoop Distributed File System (HDFS), MapReduce, and Hadoop YARN. The Hadoop MapReduce is an implementation of MapReduce programming model used for

writing applications that can process large-scale data of order of volume of multi-terabyte datasets in parallel on huge clusters in terms of thousands of nodes of commodity hardware in a safe, fault-tolerant manner. The MapReduce framework divides the input dataset into autonomous chunks that are organized by the map tasks in a parallel mode. The framework is responsible for scheduling tasks, monitoring them, and re-executing the failed tasks. YARN is added in Hadoop 2.0 framework with MapReduce programming model for cluster resource management. The YARN borrowed the resource management functions from the MapReduce model by including a separate layer to manage the resources and allowing new programming models in parallel. Currently only the MapReduce application runs on top of it. Furthermore, the intricacy to develop programs in MapReduce has led to new systems that convert programs into MapReduce, like Apache Hive [54], a data warehouse like SQL for managing and querying volumetric data, and Apache Pig [55], a platform for analyzing large datasets with a high-level language for expressing programs.

(b) *Druid*: Druid is an open-source, column-oriented, distributed analytics data store designed for business intelligence OnLine Analytical Processing (OLAP) queries on data [36, 37]. The platforms storing huge amounts of information as Hadoop do not assure about the time in which the information can be accessed or stored and also query mechanism is not efficient. The Druid addresses these issues by providing a real-time data ingestion, fast data aggregation, and flexible data exploration, and is mostly used in user-facing analytics applications. The Druid architecture mainly consists of two parts: *historical nodes*, to store and query the non-real-time information, and *real-time nodes* that can consume the stream information and respond to queries related to this data. It has other components too: *coordination nodes* for coordination of data management and data distribution on historical nodes, *brokers' nodes* to receive queries and route those queries to historical and real-time nodes, and *indexer nodes* for ingesting the real-time and batch data into the system. The real-time nodes provide real-time information and build up portions for matured information that they forward to the historical nodes that in turn hold the information in profound storage and transfer it in memory at whatever point the coordination hub needs it. It additionally uses the Apache Zookeeper, and also Hadoop segment to synchronize the components in a cluster, to deal with the present cluster state, and the MySQL for keeping the metadata related to the information sections.

(c) *ApacheHbase*: It is an open-source, non-relational, and distributed database used for performing read and write of big data randomly or in real time [38]. ApacheHbase is developed as a part of Apache Software Foundation's Apache Hadoop project modeled after the Google's Bigtable, and written in JAVA. It works on the top of HDFS having potential of providing Bigtable like capabilities for Hadoop. The data storage in ApacheHbase is done like traditional relational database management systems (RDBMS) tabular method. ApacheHbase defines a four-dimensional information model with four coordinates and defines each cell. These four coordinates are (1) Row Key which is unique per row; it does not have a data type and is treated internally as a byte

array, (2) Column Family which is unique per column and same for each row, (3) Column Qualifier defines actual columns, (4) Version is defines for each column and can have a configurable number of versions, and we can access the data for a specific version of a column qualifier. The HBase architecture is HMaster/Region Server in which the HMaster observes and manages all the Region Servers, and every Region Server serves observes and manages the underlying regions. In Hbase table, fault tolerance is provided and distributed by a basic component Aregion.

(d) *Apache Kafka*: Apache Kafka is an open-source platform for stream processing and distributed messaging created by Apache Software Foundation [39]. The goal of the project is to supply a high throughput, low latency platform for handling real-time data supply and is programmed in Scala and JAVA. Kafka is run as a cluster on one or more servers. Stream of records are stored in Kafka cluster as categories so-called *topics*. This Kafka cluster stores streams of records in classes called points. Each record comprises of a value, timestamp, and a key. Kafka has four main Application Programming Interfaces (API): (1) Producer API permits an application to distribute a surge of records to at least one Kafka subjects, (2) Consumer API permits an application to subscribe to at least one points and process the flood of records delivered to them, (3) Streams API permits an application to go about as a stream processor, expending an information stream from at least one themes and creating a yield stream to at least one yield themes, successfully changing the information streams to yield streams, (4) Connector API permits building and running reusable producers or consumers that interface Kafka themes to existing applications or information frameworks.

The Kafka is built with a publish/subscribe queue used for the streaming data. The message stream in Kafka constitutes the topics in which a producer publishes the messages and a consumer subscribes to receive them. Topics are divided for handling load balancing and fault tolerance; every division can have multiple replicas. Whenever a message is sent by a producer to Kafka, it gets saved in designated partition(s) in a disk and its copies are absorbed for fixed time duration. It allows group-consumers for load balancing ingestion information among the consumers. The producers–consumers can work synchronously or asynchronously with batch data; however, there is a trade-off between throughput and latency.

For processing cloud data in batch, distributed or real-time manner other platforms of cloud are Apache Storm, Spark Streaming, RabbitMQ [40], etc.

1.3.1.2 Cloud Infrastructure

Cloud computing primarily provides IaaS, PaaS, and SaaS services. In cloud data center IaaS is the key component that provides capabilities. The IaaS service model provides computing components with virtualization. The IaaS model provides

virtualization for computing components like storage, network platforms. Several IaaS services are discussed below:

(a) *OpenNebula*: It is an open-source cloud computing infrastructure for management of virtualized, heterogeneous data centers to enable private, public, and hybrid clouds [41, 42]. OpenNebula main goal is to provide a flexible, open, extensible, and comprehensive management layer to automatize the process of enterprise clouds by leveraging and combining existing deployed solutions for networking, monitoring or user management, storage, and virtualization. In OpenNebula all components are grouped in single key component called cloud OS. Cloud OS used to govern virtual and physical infrastructures as well as manages the allocation of virtual resources. The OpenNebula architecture is based on a front-end/host type which contains a master node and each host works as a slave node. Mater node manages and monitors the cluster and slave nodes as a common cloud platform runs virtual machines.

(b) *IaaSOpenStack*: OpenStack project started in 2010, a joint work of Rackspace Hosting and NASA. It is a free and open-supply platform deployed as an IaaS and focusing on (private or public) cloud that has an AWS EC2 like Application Programming Interface (API) [43]. The OpenNebula has a centralized system with elective components, the OpenStack incorporates a set of interrelated sub-additives with its personal APIs, that are wanted to be pre-set up and integrated for OpenStack deployment. It has a pluggable peer structure that an aspect can be installed and controlled in a different node. It provides four main helps in terms of sprint-board, compute, networking, and storage. The dashboard manages OpenStack assets and services through a customizable internet-primarily based consumer interface (UI). The pc provider is a critical factor asset that manages and controls the cloud platform. The OpenStack has storage in the form of object and block. The object storage is for redundant, fault-tolerant, and scalable facts shops, whereas the block provides chronic block degree storage for performance sensitive eventualities. The networking carrier allows network-Connectivity-as-a-service for different OpenStack offerings that offers a user interface to realize the networks, supports many technology and networking companies through a pluggable architecture.

(c) *Apache CloudStack*: It is freely available software for infrastructure cloud services and used to set up public, private, and hybrid infrastructure cloud services [44]. CloudStack, freely available software, was developed to set up and manage broad virtual machines network, as a highly scalable and available IaaS cloud computing platform. CloudStack also has an AWS EC2 support for public IaaS cloud similar to the OpenNebula and OpenStack. Its architecture is also master/slave type in which there is a CloudStack Management Server that manages all the resources in cloud. It executes in an Apache Tomcat container and many hypervisor nodes employ the virtual machines in each node via installed hypervisor. The CloudStack control Server avails the internet user interface (UI) and application program interface (API), manages storage and images, and it can be set up in a multinode mode for excessive accessibility

and load balancing among the management servers. Just like OpenNebula, it helps zones to be able to control geographically allotted nodes. A dispensed geographical zone offers a higher stage of fault tolerance as a cloud device can regain from an exterior disaster in a region. The Cloud Stack storage is of two sorts: number one and secondary garage. The former shops the disks of all virtual machines handiest and later saves the disks, snapshots, ISO snap shots, and disk templates. It does no longer aid heterogeneous secondary garage; but, it has plugin facility to make the OpenStack object garage and Amazon S3 as secondary storage. It additionally affords VDC in zones, referred to as digital private Clouds (VPC), for separation facts middle deployment.

1.3.1.3 Middleware for IoT

This IoT layer may be considered as an abstraction of functionalities and communication capabilities of the devices in IoT deployment for pervasive integration with other technologies like cloud services. Below a description of different IoT middleware like GSN, DPWS, and LinkSmart is given.

(a) *Global Sensor Networks (GSN) Middleware*: GSN middleware project is started in late 2004 at EPFL (École polytechnique fédérale de Lausanne). The GSN system is created using the sensors that may be real or virtual sensors, connected together to make the required processing path. GSN project main goal is to develop a universal platform for sensor networks and distributed processing of information produced by wireless sensor networks [45]. GSN developed in JAVA and runs on one or more systems working as the backbone of acquisition network. In GSN data streams are managed through XML specification files and live data is ingested into the system via remote wrappers. Its architecture is container based consisting of two main layers: *virtual sensor manager (VSM)* and *query manager (QM)*. VSM is used to manage virtual sensors and their infrastructure and QM layer is used to parse, execute, and planning SQL queries. For configuring alerts, VSN has a configurable notification manager and a topmost interface layer to retrieve via web services. In VMS, the connections with devices are handled by wrappers that are available for Tiny OS and other devices.

There is a big problem of data heterogeneity when we need to interpret and understand it, in spite of the fact that it supports the heterogeneous devices and accession level problems. XGSN [46] is an extension of GSN middleware; it deals with this problem by making observation and search tasks in an IoT. The XGSN provides the semantics data to virtual sensors through an expansion of the SSN ontology. XGSN regulates the annotation process about sensors, sensing devices, and their efficiency. Like GSN, the XGSN also has interfaces to manage the virtual sensors, query data, and also integration for storing and processing the stream data with the linked sensor middleware (LSM).

(b) *Device Profile for Web Services (DPWS)*: It is a large collection of WSs specifications for embedded as well as resource-constrained devices. The DPWS was developed for resource-constrained devices to make them capable of using secure web services and it is based on the service-oriented architecture (SOA). DPWS is based on many other web services specifications like WS-Addressing for advanced end point and message addressing, WS-Policy for policy exchange, WS-Security for managing security, WS-Transfer/WS-Metadata exchange for device and service description, WS-Discovery and SOAP-over-UDP for device discovery, WS-Eventing for managing subscriptions for event channels. The DPWS also partially established on W3Cs WSs standards: simple object access protocol (SOAP) 1.2, XML schema and WS-Addressing, Web Services Description Language (WSDL) 1.1; and specifies several other protocols for messaging, locating, security, and eventing. DPWS is implemented in many ways such as WS4D, implemented via C/C++ and many other via JAVA programming languages, is an open-source implementation [47]. These implementations are supported by various platforms. DPWS protocol stack is very large hence its integration with resource-constrained devices may be expensive. DPWS-compliant gateway can abstract the underlying IoT components while taking advantage of DPWS interoperability and semantics on embedded devices. LinkSmart® was initially created within the Hydra EU project for Networked Embedded Systems that can help developers to incorporate physical devices of various types into their applications through easy-to-use web services for managing any device.

(c) *LinkSmart*: It was initially designed within the Hydra EU project to build a middleware that can empower the developers to include physical devices of various types into their applications using convenient web services to manage any device. LinkSmart is based on a SOA [48]. It was designed by considering the commonly known problem of compatibility of heterogeneous devices and various protocols associated with devices.

1.3.2 CoAP and 6LowPan-Based Cloud and IoT Integration Architecture

For communication over internet between devices, the web servers (WSs) that include RESTFul and SOAP are used. SOAP WS works with exchange XML but in most of the WSs operates over HTTP; this is the main challenge for resource and energy limited devices. The CoAP, an application protocol for resource-constrained internet devices, e.g., WSN nodes and sensors, enables these devices to use the RESTful services with constrained capabilities. CoAP uses UDP rather than TCP, commonly used in HTTP, for lightweight communication between resource constraint devices. There are two main sublayers in CoAP architecture: request/response and messaging. The request/response sublayer provides communication and the messaging sublayer provides reliability and duplication of messages. CoAP provides GET, PUT, PUSH, DELETE messages requests to retrieve, create,

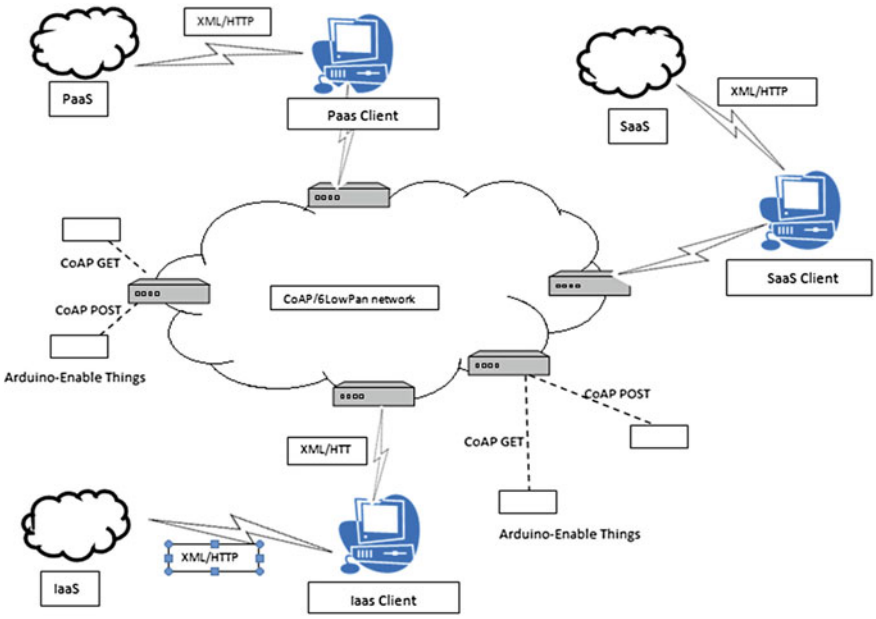

Fig. 1.4 Cloud IoT integration architecture based on CoAP and 6LowPan

update, and delete, respectively. The architecture of Cloud and IoT integration based on CoAP protocol between IoT devices, with RESTful services, and 6LoWPAN protocol for networking with things and their interaction with IaaS, PaaS, SaaS cloud computing models is shown in Fig. 1.4 [29].

1.4 Applications of Cloud Computing and IoT Integration

Integration of two rapidly growing technological areas makes sense for large number of applications [32], which is defined with characteristics, open issues, and challenges. Some applications of Cloud IoT are explained below.

1.4.1 Agriculture

IoT as a role: IoT in agriculture can be very useful as it can help in deciding more profitably crops with low cost in production [51]. The benefits of agriculture with IoT include monitoring the plants, soil, animals, controlling greenhouse environment, etc. The inputs used for agriculture like water, soil, pesticides, fertilizers, etc., along with their quantity and quality required for the crops are managed. *Cloud computing as a role*: In provincial areas, it is not viable for the farmers to

manage administration suppliers on an individual premise. They require extensive and financial administration suppliers with various administrations. With IoT, a farmer has the capacity of crop delivery straightforward to the customers not just in a little local area, but in a more extensive region. This advancement will result in better crop and effective sales of the food product and production of food products beneficial to the real world.

1.4.2 Healthcare

The physiological data of a patient is maintained by some hospitals using the sensor networks [52]. The first contribution in healthcare field is IoT and multimedia services. To empower cost effective and high-quality ubiquitous medical services, efficient smart devices and cloud services are contributing for continuous and systematic innovation in healthcare. This area requires several applications such as hospital and physician networks, laboratories, health insurance pharmaceutical companies, patients, and other entities. The healthcare application generates a large amount of data (sensor data) that requires to be maintained in a proper manner for future processing and analysis. Mobile devices can make the services more efficient for delivery of health information in future for communication and access. The common challenges in this field are security, quality of service (QoS), interoperability, and dynamicity in storage.

1.4.3 Smart City

Typical middleware for future smart city can be given through IoT, obtaining information from sensing infrastructure, IoT technology (RFID sensors and geotagging), and putting data in a consistent manner, to strengthen the discovery, integration, and interconnection of actuators and sensors. This frames easy to applications for smart cities that are real-time and widespread connectivity [13]. This makes easier to third parties to develop IoT plugin to be connected to Cloud. The familiar issues are associated with security, real-time interaction, and resilience.

1.4.4 Smart Home and Smart Metering

To empower automation of regular in-home activities, the IoT has large applications for home atmosphere wherein the embedded devices have been used [53]. To build flexible applications with less code lines and to handle complex tasks, cloud is the best source to manage even huge data. When a single family smart home accessing reusable service is to be accessed online, some of the requirements should

be satisfied: automation (home-based application should be attached to service provider like smart home-based Cloud), internal network inter connection (each intelligent digital thing in the home should interact with each other), intelligent remote control (smart home devices or objects can be managed or intelligently operated from anywhere). The existing literatures discuss metered solutions to provide identification of appliances, wireless sensor networks, and intelligent management of heating, lighting, consumption of energy, and air conditioning.

1.4.5 Video Surveillance

Video surveillance is one of the most important intelligent things as a part of security related issue because it works as a monitoring and self-management system [49]. The complex video analysis requires cloud-based solutions to fulfill the requirement of storage and processing. The video surveillance helps identifying, storing, and managing the video information from a camera and data delivery efficiently to a number of users through internet, load balancing, and fault-tolerance fashions.

1.5 Issues and Challenges for Integration

Some of the open issues in Cloud and IoT integration require further investigation that may be future directions [8, 33].

1.5.1 Security and Privacy Issues

Security and privacy are the main challenges to set up an IoT infrastructure. The devices in IoT are resource constrained, which can be exposed to attacks and threats. IoT devices sometimes use as well as generate sensitive information such as private information or vital infrastructure, which needs privacy to be included with devices, IoT network, and cloud infrastructure. The security and privacy aspects in applications of IoT in real world are discussed in [50]. To make IoT infrastructure secure several security techniques should be integrated with IoT. These security techniques involve protocol and network security for communication of heterogeneous devices of IoT, identity management for these heterogeneous devices, privacy for big data generated by IoT entities, trust and governance in IoT systems for communication and common framework, fault tolerance for attacks, and failure of IoT systems. The threats models, of an attacker, in the IoT are eavesdropping, node capture to gather data, denial of service (DoS), physical damage, and controlling various entities. Furthermore, Cloud computing acquisition in IoT also has security and privacy concerns.

1.5.2 Ipv6

One of the main components of IoT is internet which has address limitations due to IPv4 addressing scheme. The technologies for resource-constrained devices like CoAP can interact directly with embedded devices via Internet. Continuous growth of these technologies requires elimination of the network address translation (NAT) mechanisms so that they can address each IoT device or service with a unique IP address. The IPv6 addresses this problem using 128 bit IP address and also has several advantages like the increased range of devices connected to internet, source to destination connectivity, and an agreement with open REST interfaces [49]. In embedded devices of IoT, implementation of the IPv6 can be done by employing 6LoWPAN and ZigBee IP specifications, which still needs to be spread in many commercial platforms. The IoT network is trending towards networks of human-to-machine and machine-to-machine communications using IPv6 from networks with human initiated actions.

1.5.3 Need for Standards

In cloud and IoT paradigm, necessary protocols, architectures are required that is being standardized by the scientific community. It interconnects the enhanced services and heterogeneous smart objects to realize Cloud and IoT integration paradigm [23]. The important paradigm is Mobile-To-Mobile (M2M) with a little standard. So, the available solutions use internet, standard web, and cellular technologies. Most of the architectures at the primary phase of IoT are either from cloud or from wireless sensor networks.

1.5.4 Complex Data Mining

The issues related to big data cannot be addressed by the existing technologies. When high magnitude big data is generated, its high-level frequency, gap between the data availability, and its organization for processing get wider. Further research is required to address the challenges of big data, heterogeneous spatiotemporal (geo-related and sparsely distributed) data, i.e., the data that worth more mixed with erroneous data, are not directly consumable using virtualization platform in IoT. To create interesting and easy to perceive visualization, new methods are to be developed (e.g., 3D, geographic information system (GIS)).

1.5.5 Cloud Capabilities

In a networked environment, the security is one of the major issues for Cloud and IoT integration due to various attacks on both of IoT (i.e., RFID, WSN) and Cloud side. Integrity, confidentiality, and authenticity in IoT can be done using encryption that can address inside attacks and also implementable on processing or embedded devices. The RFID components achieve high-level security due to high-level intelligence. The QoS requirements of diverse users, seamless execution of applications, and domain specific programming tools are also required to deliver reliable services. Duplication for cloud scheduling algorithms can help in failure management.

1.5.6 Fog Computing

The next step to cloud computing is fog computing which is an intermediate between the edge of the network and Cloud to deal with latency-sensitive applications to meet their delay requirements [50]. Alike to cloud, the Fog services include computing, data storage, application services to target users. Future work in smart grid will develop Fog computing paradigm. Even Fog devices are being developed to interact directly to a Cloud.

1.5.7 Energy Efficiency

With the ubiquity of sensor networks and their availability with the cloud, this will unavoidably prompt a considerable measure of information correspondences, which devours a great deal of power. A commonplace wireless sensor hub is made out of four parts: sensing unit, processing unit, transceiver, and power unit. If there should be an occurrence of video sensing, video encoding what's more, translating, power assumes a key part. Typically, video encoding is more complex, as contrasted with interpreting. The purpose for this is for productive pressure, the encoder needs to dissect the repetition in the video. It won't be reasonable to have a transitory power supply, similar to batteries and need to supplant them from time to time. With billions of sensors and low power devices, it is past probability. Having effective use of energy and rather perpetual power supply would be required. There ought to be means for sensors to produce power from the condition, as sun oriented energy, vibration, and air. Moreover, successful rest mode can be exceptionally convenient in such manner also. Another arrangement exhibited in is bringing cloud assets privately, known as Fog computing. Mist alludes to a limited cloud, which can be utilized for process offloading reason for the fundamental IoT devices.

1.6 Conclusion and Future Scope

In cloud computing and IoT model, the IoT is both dynamic and universal networked infrastructure oriented, and manages self-configuring nodes (things) with high intelligence. Since the IoT has limited capabilities in terms of processing potential and storage, the issues related to performance, security, reliability, privacy are of major concern. The integration of IoT with Cloud Computing is more beneficial for the applications requiring unlimited capabilities such as storage and processing capability. An ample number of applications such as smart city, healthcare, smart home, smart metering, video surveillance, agriculture, and automotive can significantly be improved in CloudIoT or Cloud of Things paradigm. High-level architecture based on Cloud Assisted Computing manages the complexity of smart object-based Internet of Things. Since the cloud and IoT integration is in infancy stage, no standard architecture is available. We have presented major key challenges in CloudIoT system. Working on these challenges would contribute in standardizing the CloudIoT.

In order to standardize and to come up with the full potential of Cloud and IoT integration, further research efforts are required in multiple directions as mentioned.

- For identifying, naming, and addressing things in IoT, it's far required to guide huge range of IoT devices and mobility of them. Though the IPv6 can be an efficient answer, yet to accumulate it on a big scale is an ongoing work. Additional research on IPv6, for IoT infrastructure, is necessary to both speed up this slow process in specific scenarios (e.g. access networks) and to cope with new mobility and scalability requirements.
- For detecting environmental adjustments, the solutions primarily depend upon IoT information will permit the delivery of context-based totally offerings and offer the best offerings relying at the state of affairs. Such possibility will inspire the research for growing extra effective algorithms to supply the personalized contents and classified ads.
- Large scale aid for multi-networking including multihoming, connection handover, and roaming will be obligatory for enhancing the community reliability and making sure redundancy, continuous connectivity, QoS, and fault tolerance. In this context, the solutions based totally on software described networking are also envisaged.
- Many applications of Cloud and IoT integration will take advantage of efficient and adaptable mechanisms in designing logically isolated network parts over the global network infrastructures, which can be a crucial driving force for research in virtualization and software-described networking fields.

References

1. Devipriya S. Contribution of internet of things: a survey. J Web Develop Web Des. 2016;1(3):1–6.
2. Gubbi J, Buyya R, Marusic S, Palaniswami M. Internet of Things (IoT): a vision, architectural elements, and future directions. Futur Gener Comput Syst. 2013;29(7):1645–60.
3. Mell P, Grance T. The NIST definition of cloud computing. 2011.
4. Hamdaqa M, Livogiannis T, Tahvildari L. A reference model for developing cloud applications. In: CLOSER 2011—Proceeding of the 1st international conference on could computing and services science; 2011. p. 98–103.
5. Sheng Z, Wang H, Yin C, Hu X, Yang S, Leung VC. Lightweight management of resource-constrained sensor devices in internet of things. IEEE Internet Things J. 2015;2(5):402–11.
6. Aitken R, Chandra V, Myers J, Sandhu B, Shifren L, Yeric G. Device and technology implications of the internet of things. In: 2014 symposium on VLSI technology (VLSI-technology): digest of technical papers. IEEE; 2014. p. 1–4.
7. Sheng Z, Yang S, Yu Y, Vasilakos A, Mccann J, Leung K. A survey on the ietf protocol suite for the internet of things: standards, challenges, and opportunities. IEEE Wirel Commun. 2013;20(6):91–8.
8. Botta A, De Donato W, Persico V, Pescapé A. Integration of cloud computing and internet of things: a survey. Futur Gener Comput Syst. 2016;56:684–700.
9. Lee K, Murray D, Hughes D, Joosen W. Extending sensor networks into the cloud using amazon web services. In: 2010 IEEE international conference on networked embedded systems for enterprise applications (NESEA). IEEE; 2010. p. 1–7.
10. Botta A, De Donato W, Persico V, Pescapé A. On the integration of cloud computing and internet of things. In: 2014 international conference on future internet of things and cloud (FiCloud). IEEE; 2014. p. 23–30.
11. Alhakbani N, Hassan MM, Hossain MA, Alnuem M. A framework of adaptive interaction support in cloud-based internet of things (IoT) environment. In: international conference on internet and distributed computing systems. Springer International Publishing; 2014. p. 136–146.
12. Dash SK, Mohapatra S, Pattnaik PK. A survey on applications of wireless sensor network using cloud computing. Int J Comput Sci Eng Technol. 2010;1(4):50–5. E-ISSN: 2044-6004
13. Suciu G, Vulpe A, Halunga S, Fratu O, Todoran G, Suciu, V. Smart cities built on resilient cloud computing and secure internet of things. In: 2013 19th international conference on control systems and computer science (CSCS). IEEE; 2013. p. 513–518.
14. Rao BP, Saluia P, Sharma N, Mittal A, Sharma SV. Cloud computing for internet of things and sensing based applications. In: 2012 sixth international conference on sensing technology (ICST). IEEE; 2012. p. 374–380.
15. Zaslavsky A, Perera C, Georgakopoulos D. Sensing as a service and big data. 2013. preprint arXiv:1301.0159.
16. Prati A, Vezzani R, Fornaciari M, Cucchiara R. Intelligent video surveillance as a service. In: Intelligent multimedia surveillance. Berlin, Heidelberg: Springer; 2013. p. 1–16.
17. Guinard D, Floerkemeier C, Sarma S. Cloud computing, REST and mashups to simplify RFID application development and deployment. In: Proceedings of the second international workshop on web of things. ACM; 2011. p. 9.
18. Nimbits platform. http://www.nimbits.com. Accessed 10 Mar 2017.
19. Thingspeak platform. https://www.thingspeak.com. Accessed 10 Mar 2017.
20. Paraimpu. http://paraimpu.crs4.it. Accessed 10 Mar 2017.
21. Device cloud. http://www.idigi.com/devicecloud. Accessed 10 Mar 2017.
22. SensorCloud. http://www.sensorcloud.com.Accessed 10 Mar 2017.
23. Arduino. http://www.arduino.cc. Accessed 10 Mar 2017.
24. Stack4Things. http://stack4things.unime.it. Accessed 10 Mar 2017.
25. Wiring. http://wiring.org.co. Accessed 10 Mar 2017.

26. Sunspot. http://www.sunspotworld.com. Accessed 10 Mar 2017.
27. Mbed. http://mbed.org. Accessed 10 Mar 2017.
28. Hou L, Zhao S, Xiong X, Zheng K, Chatzimisios P, Hossain MS, Xiang W. Internet of things cloud: architecture and implementation. 2016. preprint arXiv:1609.07712.
29. Zhou J, Leppanen T, Harjula E, Ylianttila M, Ojala T, Yu C, Yang LT. Cloudthings: a common architecture for integrating the internet of things with cloud computing. In: 2013 IEEE 17th international conference on computer supported cooperative work in design (CSCWD). IEEE; 2013. p. 651–657.
30. Zhou J, Leppanen T, Harjula E, Ylianttila M, Ojala T, Yu C, Yang LT. Cloudthings: a common architecture for integrating the internet of things with cloud computing. In: 2013 IEEE 17th international conference on computer supported cooperative work in design (CSCWD). IEEE; 2013. p. 651–657.
31. Kumarage H, Ibrahim K, Abdulatif A, Zahir T, Xun Y. Secure data analytics for cloud-integrated internet of things applications. IEEE Cloud Comput. 2016;3(2):46–56.
32. Addo ID, Ahamed SI, Yau SS, Buduru A. Reference architectures for privacy preservation in cloud-based IoT applications. IJSC. 2014;2(4)
33. Díaz M, Martín C, Rubio B. State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. J Netw Comput Appl. 2016;67:99–117.
34. Shvachko K, Kuang H, Radia S, Chansler R. The hadoop distributed file system. In: 2010 IEEE 26th symposium on mass storage systems and technologies (MSST). IEEE; 2010. p. 1–10.
35. Hadoop, http://hadoop.apache.org. Accessed 10 Mar 2017.
36. Yang F, Tschetter E, Léauté X, Ray N, Merlino G, Ganguli D. Druid: a real-time analytical data store. In: Proceedings of the 2014 ACM SIGMOD international conference on management of data. ACM; 2014. p. 157–168.
37. Druid, http://druid.io. Accessed 10 Mar 2017.
38. Apache HBase, http://hbase.apache.org. Accessed 10 Mar 2017.
39. Apache Kafka, http://kafka.apache.org. Accessed 10 Mar 2017.
40. RabbitMQ, https://www.rabbitmq.com. Accessed 10 Mar 2017.
41. Moreno-Vozmediano R, Montero RS, Llorente IM. Iaas cloud architecture: from virtualized datacenters to federated cloud infrastructures. Computer. 2012;45(12):65–72.
42. OpenNebula, http://opennebula.org. Accessed 10 Mar 2017.
43. IaaSOpenStack, https://www.openstack.org. Accessed 10 Mar 2017.
44. Apache CloudStack, http://cloudstack.apache.org. Accessed 10 Mar 2017.
45. GSN, https://github.com/LSIR/gsn/wiki. Accessed 10 Mar 2017.
46. Calbimonte JP, Sarni S, Eberle J, Aberer K. XGSN: an open-source semantic sensing middleware for the web of things. In: TC/SSN@ ISWC; 2014. p. 51–66.
47. WSD4, http://ws4d.org. Accessed 10 Mar 2017.
48. LinkSmart, https://linksmart.eu. Accessed 10 Mar 2017.
49. Ziegler S, Crettaz C, Thomas I. IPv6 as a global addressing scheme and integrator for the Internet of Things and the cloud. In: 2014 28th international conference on advanced information networking and applications workshops (WAINA). IEEE; 2014. p. 797–802.
50. Roman R, Zhou J, Lopez J. On the features and challenges of security and privacy in distributed internet of things. Comput Netw. 2013;57(10):2266–79.
51. Biduaa KR, Patela CN. Internet of things and cloud computing for agriculture in India. International Journal of Innovative and Emerging Research in Engineering. 2015;2(12): 2394–3343.
52. Doukas C, Maglogiannis I. Bringing IoT and cloud computing towards pervasive healthcare. In: Sixth IEEE international conference on innovative mobile and internet services in ubiquitous computing (IMIS), Palermo, Italy, July 4–6 2012;2012. p. 922–926.
53. Soliman M, Abiodun T, Hamouda T, Zhou J, Lung CH. Smart home: Integrating internet of things with web services and cloud computing. In: 5th IEEE international conference on cloud computing technology and science (CloudCom), December 2013, vol. 2;2013. p. 317–320.
54. Apache Hive. 2017. https://hive.apache.org. Accessed 10 Mar 2017.
55. Apache Pig. 2017. https://pig.apache.org. Accessed 10 Mar 2017.