MODULE 4: ASSOCIATED IOT TECHNOLOGIES

CLOUD COMPUTING

➢ INTRODUCTION

Sensor nodes are the key components of Internet of Things (IoT). These nodes are resource-constrained in terms of storage, processing, and energy. Moreover, in IoT, the devices are connected and communicate with one another by sharing the sensed and processed data. Handling the enormous data generated by this large number of heterogeneous devices is a non-trivial task. Consequently, cloud computing becomes an essential building block of the IoT architecture. It aims at providing an extensive overview of cloud computing. Additionally, check yourself will help the learner to learn different concepts are related to cloud computing. Cloud computing is more than traditional network computing. Unlike network computing, cloud computing comprises a pool of multiple resources such as servers, storage, and network from single/multiple organizations. These resources are allocated to the end users as per requirement, on a payment basis. In cloud computing architecture, an end user can request for customized resources such as storage space, RAM, operating systems, and other software to a cloud service provider (CSP) as shown in Figure 4.1. For example, a user can request for a Linux operating system for running an application from a CSP; another end user can request for Windows 10 operating system from the same CSP for executing some application. The cloud services are accessible from anywhere and at any time by an authorized user through Internet connectivity.

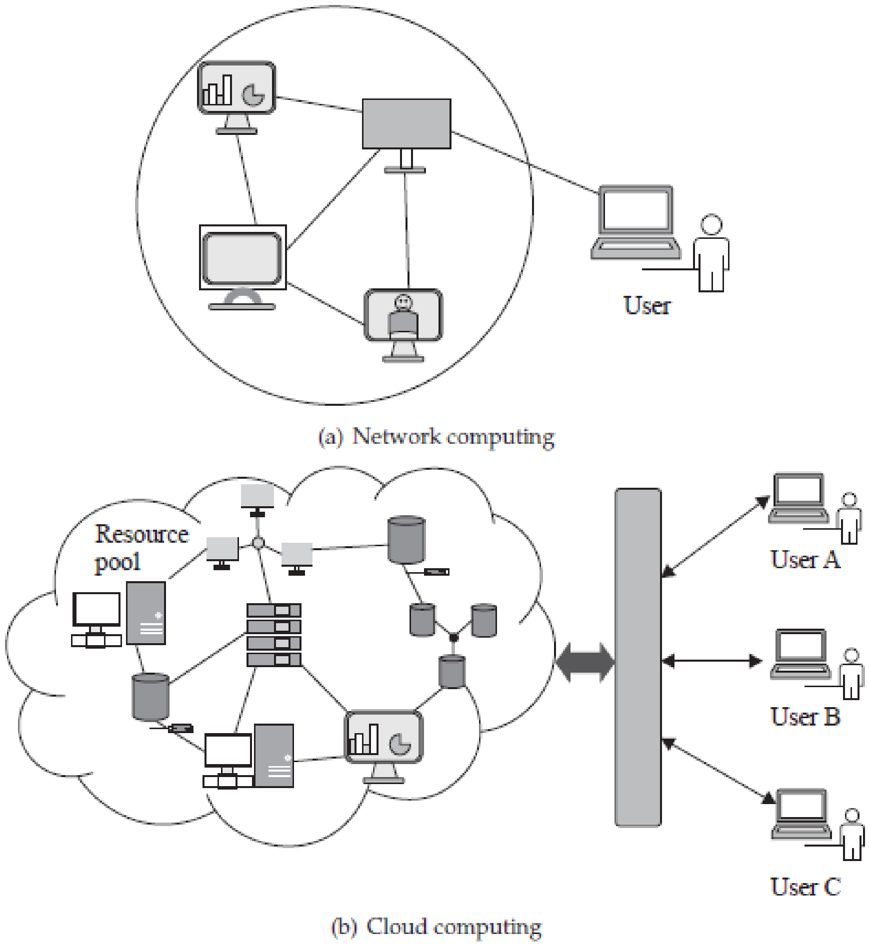


Figure 4.1 Network computing versus cloud computing

➢ VIRTUALIZATION

The key concept of cloud computing is virtualization. The technique of sharing a single resource among multiple end user organizations or end users is known as virtualization. In the virtualization process, a physical resource is logically distributed among multiple users. However, a user perceives that the resource is unlimited and is dedicatedly provided to him/her. Figure 4.2(a) represents a traditional desktop, where an application (App) is running on top of an OS, and resources are utilized only for that particular application. On the other hand, multiple resources can be used by different end users through virtualization software, as shown in Figure 4.2(b).Virtualization software separates the resources logically so that there is no conflict among the users during resource utilization.

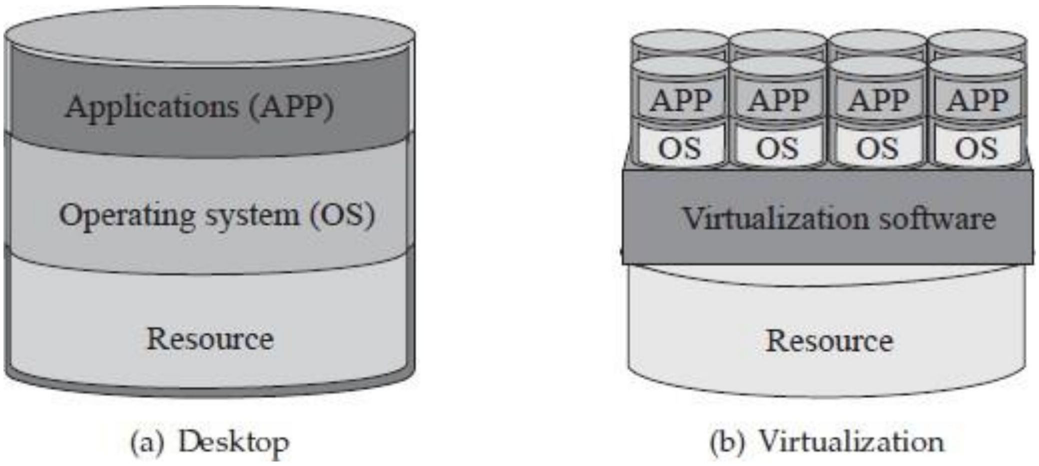


Figure 4.2 Traditional desktop versus virtualization

Advantages of virtualization

With the increasing number of interconnected heterogeneous devices in IoT, the importance of virtualization also increases. In IoT, a user is least bothered about where the data from different heterogeneous devices are stored or processed for a particular application. Users are mainly concerned for their services. Typically, there are different software such as VMware, which enable the concept of virtualization. With the increasing importance of cloud computing, different organizations and individuals are using it extensively. Moreover, there is always a risk of system crash at any instant of time. In such a scenario, cloud computing plays a vital role by keeping backups through virtualization. Primarily, there are two entities in a cloud computing architecture: end users and CSP. Both end users and CSP are benefited in several aspects through the process of virtualization. The major advantages, from the perspective of the end user and CSP, are as follows:

Advantages for End Users

- **Variety:** The process of virtualization in cloud computing enables an end user organization to use various types of applications based on the requirements. As an example, suppose John takes up still photography as a hobby. His resource-limited PC can barely handle the requirements for a photo editing software, say X-photoeditor. In order to augment his PC's regular performance, he uninstalls the X-photoeditor software and purchases a cloud service, which lets him access a virtual machine (VM). In his VM, he installs the X-photoeditor software, by which he can edit photos efficiently and, most importantly, without worrying about burdening his PC or running out of processing resources. After six months, John's interest in his hobby grows and he moves on to video-editing too. For editing his captured videos, he installs a video editing software, Y-video editor, in his VM and can edit videos efficiently. Additionally, he has the option of installing and using a variety of software for different purposes.

- **Availability:** Virtualization creates a logical separation of the resources of multiple entities without any intervention of end users. Consequently, the concept of virtualization makes available a considerable amount of resources as per user requirements. The end users feel that there are unlimited resources present dedicatedly for him/her. Let us suppose that Jane uses a particular email service. Her account has been active for over ten years now; however, it offers limited storage of 2 GB. Due to the ever accumulating file attachments in different emails, her 2 GB complimentary space is exhausted. However, there is a provision that if she pays $100 annually, she can attach additional space to her mail service. This upgrade allows her to have more storage at her disposal for a considerable time in the future.
- **Portability:** Portability signifies the availability of cloud computing services from anywhere in the world, at any instant of time. For example, a person flying from the US to the UK still has access to their documents, although they cannot physically access the devices on which the data is stored. This has been made possible by platforms such as Google Drive.
- **Elasticity:** Through the concept of virtualization, an end user can scale-up or scale-down resource utilization as per requirements. We have already explained that cloud computing is based on a pay-per-use model. The end user needs to pay the amount based on their usage. For example, Jack rents two VMs in a cloud computing infrastructure from a CSP. VM1 has the Ubuntu operating system (OS), on which Jack is simulating a network scenario using Network Simulator-2 (NS2). VM2 has Windows 10 OS, on which he is running a MATLAB simulation. However, after a few days, Jack feels that his VM2 has served its purpose and is no longer required. Consequently, he releases VM2 and, after that, he is only billed for VM1. Thus, Jack can scale-up or scale-down his resources in cloud computing, which employs the concept of virtualization.

Advantages for CSP

- **Resource Utilization:** Typically, a CSP in a cloud computing architecture procures resources on their own or get them from third parties. These resources are distributed among different users dynamically as per their requirements. A segment of a particular resource provided to a user at a time instant, can be provided to another user at a different time instant. Thus, in the cloud computing architecture, resources can be re-utilized for multiple users.
- **Effective Revenue Generation:** A CSP generates revenue from the end users based on resource utilization. As an example, today, a user **A** is utilizing storage facility from a particular CSP. The user will release the storage after a few days when his/her requirement is complete. The CSP earns some revenue from user A for the utilization of the allocated storage facility. In the future, the CSP can provide the same storage facility to a different user, **B**. Again, the CSP can generate revenue from user B for his/her storage utilization.

Types of virtualization

Based on the requirements of the users, we categorized virtualization as shown in Figure 4.3.

- **Hardware Virtualization:** This type of virtualization indicates the sharing of hardware resources among multiple users. For example, a single processor appears as many different processors in a cloud computing architecture. Different operating systems can be installed in these processors and each of them can work as stand-alone machines.

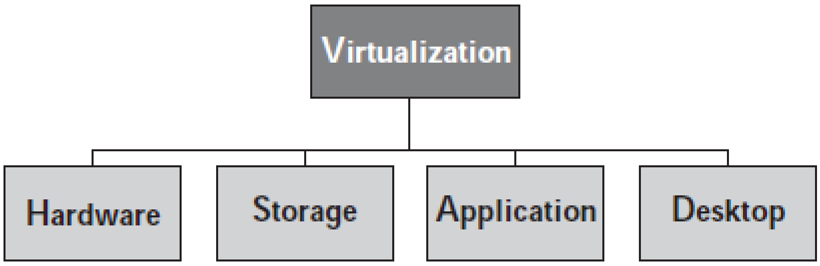


Figure 4.3 Types of virtualization

- **Storage Virtualization:** In storage virtualization, the storage space from different entities are accumulated virtually, and seem like a single storage location. Through storage virtualization, a user's documents or files exist in different locations in a distributed fashion. However, the users are under the impression that they have a single dedicated storage space provided to them.
- **Application Virtualization:** A single application is stored at the cloud end. However, as per requirement, a user can use the application in his/her local computer without ever actually installing the application. Similar to storage virtualization, in application virtualization, the users get the impression that applications are stored and executed in their local computer.
- **Desktop Virtualization:** This type of virtualization allows a user to access and utilize the services of a desktop that resides at the cloud. The users can use the desktop from their local desktop.

➢ CLOUD MODELS

As per the National Institute of Standards and Technology (NIST) and Cloud Computing Standards Roadmap Working Group, the cloud model can be divided into two parts: (1) Service model and (2) Deployment model as shown in Figure 4.4. Further the service model is categorized as: Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS). On the other hand, the deployment model is further categorized as: Private cloud, Community cloud, Public cloud, and Hybrid cloud.

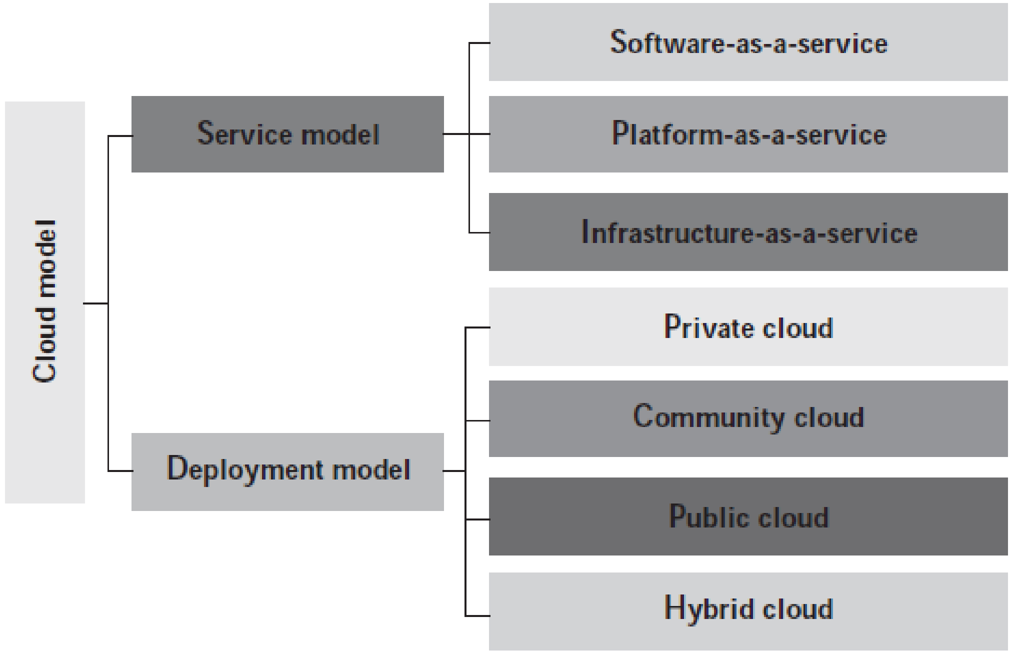


Figure 4.4 Cloud model

Service Model

The service model is depicted in Figure 4.5.

- **Software-as-a-Service (SaaS):** This service provides access to different software applications to an end user through Internet connectivity. For accessing the service, a user does not need to purchase and install the software applications on his/her local desktop. The software is located in a cloud server, from where the services are provided to multiple end users. SaaS offers scalability, by which users have the provision to use multiple software applications as per their requirements. Additionally, a user does not need to worry about the update of the software applications. These software are accessible from any location. One example of SaaS is Microsoft Office 365.
- **Platform-as-a-Service (PaaS):** PaaS provides a computing platform, by which a user can develop and run different applications. The cloud user need not go through the burden of installing and managing the infrastructure such as operating system, storage, and networks. However, the users can develop and manage the applications that are running on top of it. An example of PaaS is Google App Engine.
- **Infrastructure-as-a-Service (IaaS):** IaaS provides infrastructure such as storage, networks, and computing resources. A user uses the infrastructure without purchasing the software and other network components. In the infrastructure provided by a CSP, a user can use any composition of the operating system and software. An example of IaaS is Google Compute Engine.

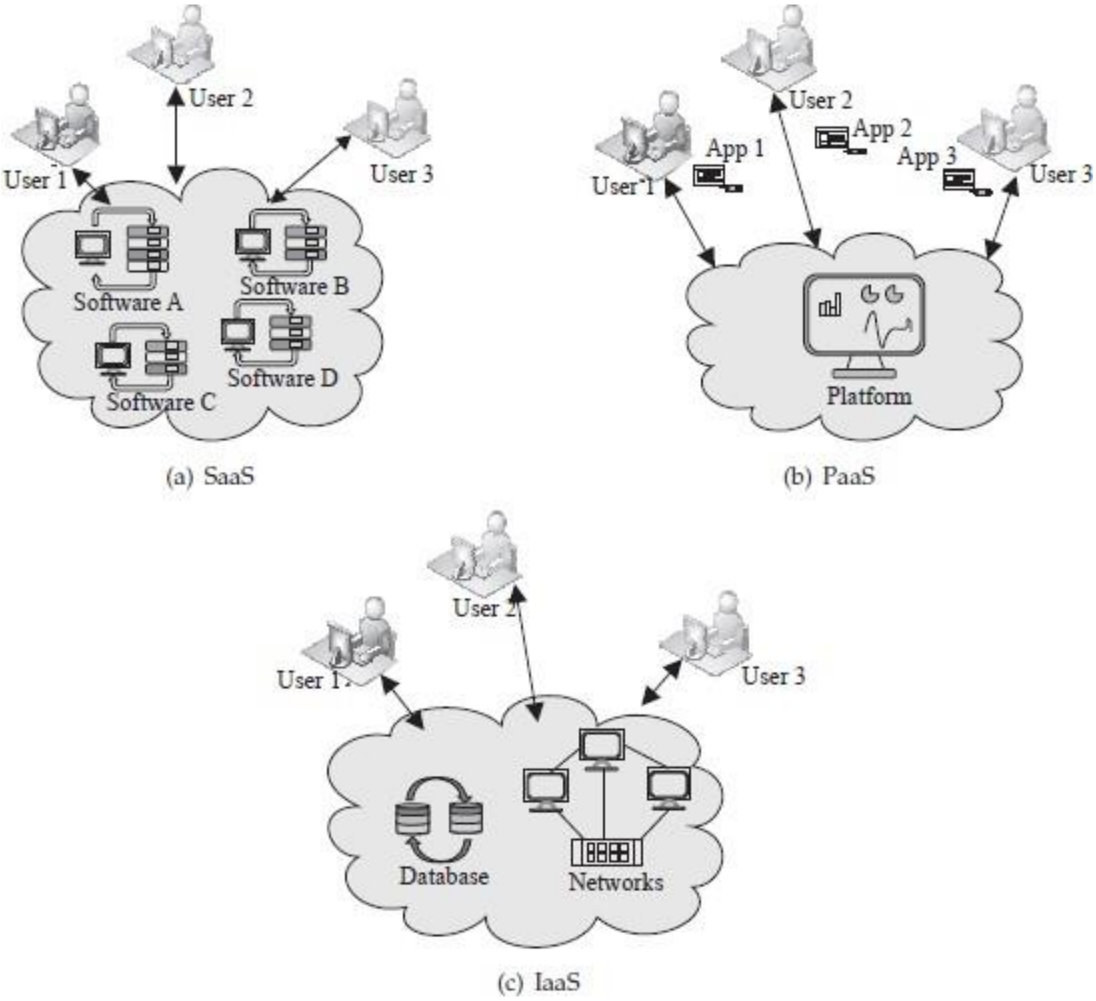


Figure 4.5 Service models

Deployment Model

- **Private Cloud:** This type of cloud is owned explicitly by an end user organization. The internal resources of the organization maintain the private cloud.
- **Community Cloud:** This cloud forms with the collaboration of a set of organizations for a specific community. For a community cloud, each organization has some shared interests.
- **Public Cloud:** The public cloud is owned by a third party organization, which provides services to the common public. The service of this cloud is available for any user, on a payment basis.
- **Hybrid Cloud:** This type of cloud comprises two or more clouds (private, public, or community).

➢ SERVICE-LEVEL AGREEMENT IN CLOUD COMPUTING

The most important actors in cloud computing are the end user/customer and CSP. Cloud computing architecture aims to provide optimal and efficient services to the end users and generate revenue from them as per their usage. Therefore, for a clear understanding between CSP and the customer about the services, an agreement is required to be made, which is known as service-level agreement (SLA). An SLA provides a detailed description of the services that will be received by the customer. Based on the SLA, a customer can be aware of each and every term and condition of the services before availing them. An SLA may include multiple organizations for making the legal contract with the customers.

Importance of SLA

An SLA is essential in cloud computing architecture for both CSP and customers. It is important because of the following reasons:

- **Customer Point of View**: Each CSP has its SLA, which contains a detailed description of the services. If a customer wants to use a cloud service, he/she can compare the SLAs of different organizations. Therefore, a customer can choose a preferred CSP based on the SLAs.
- **CSP Point of View**: In many cases, certain performance issues may occur for a particular service, because of which a CSP may not be able to provide the services efficiently. Thus, in such a situation, a CSP can explicitly mention in the SLA that they are not responsible for inefficient service.

Metrics for SLA

Depending on the type of services, an SLA is constructed with different metrics. However, a few common metrics that are required to be included for constructing an SLA are as follows:

- **Availability:** This metric signifies the amount of time the service will be accessible for the customer.
- **Response Time:** The maximum time that will be taken for responding to a customer request is measured by response time.
- **Portability:** This metric indicates the flexibility of transferring the data to another service.
- **Problem Reporting:** How to report a problem, whom and how to be contacted, is explained in this metric.
- **Penalty:** The penalty for not meeting the promises mentioned in the SLA.

➢ CLOUD IMPLEMENTATION

Cloud simulation

With the rapid deployment of IoT infrastructure for different applications, the requirement for cloud computing is also increasing. It is challenging to estimate the performance of an IoT system with the cloud before real implementation. On the other hand, real deployment of the cloud is a complex and costly procedure. Thus, there is a requirement for simulating the system through a cloud simulator before real implementation. There are many cloud simulators that provide pre-deployment test services for repeatable performance evaluation of a system. Typically, a cloud simulator provides the following advantages to a customer:

- Pre-deployment test before real implementation
- System testing at no cost
- Repeatable evaluation of the system
- Pre-detection of issues that may affect the system performance
- Flexibility to control the environment

Currently, different types of cloud simulators are available. A few cloud simulators are listed here:

CloudSim

Description: CloudSim is a popular cloud simulator that was developed at the University of Melbourne. This simulator is written in a Java-based environment. In CloudSim, a user is allowed to add or remove resources dynamically during the simulation and evaluate the performance of the scenario.

Features: CloudSim has different features, which are listed as follows:

- The CloudSim simulator provides various cloud computing data centers along with different data center network topologies in a simulation environment.
- Using CloudSim, virtualization of server hosts can be done in a simulation.
- A user is able to allocate virtual machines (VMs) dynamically.
- It allows users to define their own policies for the allocation of host resources to VMs.
- It provides flexibility to add or remove simulation components dynamically.
- A user can stop and resume the simulation at any instant of time.

CloudAnalyst

Description: CloudAnalyst is based on CloudSim. This simulator provides a graphical user interface (GUI) for simulating a cloud environment, easily. The CloudAnalyst is used for simulating large-scale cloud applications.

Features:

- The CloudAnalyst simulator is easy to use due to the presence of the GUI.
- It allows a user to add components and provides a flexible and high level of configuration.
- A user can perform repeated experiments, considering different parameter values.
- It can provide a graphical output, including a chart and table.

GreenCloud

Description: GreenCloud is developed as an extension of a packet level network simulator, NS2. This simulator can monitor the energy consumption of different network components such as servers and switches.

Features:

- GreenCloud is an open-source simulator with user-friendly GUI.
- It provides the facility for monitoring the energy consumption of the network and its various components.
- It supports the simulations of cloud network components.
- It enables improved power management schemes.
- It allows a user to manage and configure devices, dynamically, in simulation.

An open-source cloud: OpenStack

For the real implementation of cloud, there are various open-source cloud platforms available such as OpenStack, CloudStack, and Eucalyptus. Here, we will discuss the OpenStack platform briefly. The OpenStack is free software, which provides a cloud IaaS to users. A user can easily use this cloud with the help of a GUI-based web interface or through the command line. OpenStack supports a vastly scalable cloud system, in which different pre-configured software suites are available. The service components of OpenStack along with their functions are depicted in Table 4.1.

Features of OpenStack

- OpenStack allows a user to create and deploy virtual machines.
- It provides the flexibility of setting up a cloud management environment.
- OpenStack supports an easy horizontal scaling: dynamic addition or removal of instances for providing services to multiple numbers of users.
- This cloud platform allows the users to access the source code and share their code to the community.

Table 4.1 Components in OpenStack

Component	Function
Nova	Compute
Neutron	Networking
Cinder	Block storage
Keystone	Identity
Glance	Image
Swift	Object storage
Horizon	Dashboard
Trove	Database
Sahara	Elasticmap reduce
Manila	Shared file system
Designate	DNS
Searchlight	Search
Barbican	Key manager

A commercial cloud: Amazon web services (AWS)

Besides the open-source cloud, there are various commercial cloud infrastructures available in the market. Few of the popular commercial cloud infrastructures are Amazon Web Services (AWS), Microsoft Azure, and Google App Engine. In this section, we will discuss the different features of AWS. A user can launch and manage server instances in AWS. Typically, a web interface is used to handle the instances. Additionally, AWS provides different APIs (application programming interfaces), tools, and utilities for users. Like other commercial clouds, Amazon AWS follows the pay-per-use model. This cloud infrastructure provides a virtual computing environment, where different configurations, such as CPU, memory, storage, and networking capacity are available.

Features of AWS

- It provides flexibility to scale and manage the server capacity.
- AWS provides control to OS and deployment software.
- It follows the pay-per-use model.
- The cloud allows a user to establish connectivity between the physical network and private virtual network.
- The developer tools in this cloud infrastructure help a user for fast development and deployment of the software.
- AWS provides excellent management tools, which help a user to monitor and automate different components of the cloud.
- The cloud provides machine learning facilities, which are very useful for data scientists and developers.

- For extracting meaning from data, analytics play an important role. AWS also provides a data analytics platform.

➢ SENSOR-CLOUD: SENSORS-AS-A-SERVICE

The new concept known as Sensors-as-a-Service (Se-aaS) in a sensor-cloud architecture is explored. Virtualization of resources is the backbone of cloud computing. Similarly, in a sensor-cloud, virtualization of sensors plays an essential role in providing services to multiple users. Typically, in a sensor-cloud architecture, multiple users receive services from different a sensor nodes, simultaneously. However, the users remain oblivious to the fact that a set of sensor nodes is not dedicated solely to them for their application requirements. In reality, a particular sensor may be used for serving multiple user applications, simultaneously. The main aim of sensor-cloud infrastructure is to provide an opportunity for the common mass to use Wireless Sensor Networks (WSNs) on a payment basis. Similar to cloud computing, sensor-cloud architecture also follows the pay-per-use model.

Importance of sensor-cloud

The sensor-cloud infrastructure is based on the concept of cloud computing, in which a user application is served by a set of homogeneous or heterogeneous sensor nodes. These sensor nodes are selected from a common pool of sensor nodes, as per the requirement of user applications. Using the sensor-cloud infrastructure, a user receives data for an application from multiple sensor nodes without owning them. Unlike sensor-cloud, if a user wants to use traditional WSN for a certain application, he/she has to go through different pre-deployment and post-deployment hurdles.

Figures 4.6 depicts the usage of sensor nodes using traditional WSN and sensor cloud infrastructure. With the help of a case study, we will discuss the advantages of sensor-cloud over traditional WSN.

Case Study: John is a farmer, and he has a significantly vast farmable area with him. As manual supervision of the entire field is very difficult, he has planned to deploy a WSN in his farming field. Before purchasing the WSN, he has to decide which sensors should be used in his fields for sensing the different agricultural parameters. Additionally, he has to decide the type and number of other components such as an electronics circuit board and communication module required along with the sensors. As there are numerous vendors, it is challenging for him to choose the correct (in terms of quality and cost) vendor, as well as the sensor owner from whom the WSN will be procured. He finally decides the type of sensors along with the other components that are required for monitoring his agricultural field. Now, John faces the difficulty of optimally planning the sensor node deployment in his fields. After going through these hurdles, he decides on the number of sensor nodes that are required for monitoring his field. Finally, John procures the WSNs from a vendor. After procurement, he deploys the sensor nodes and connects different components. As WSN consists of different electronic components, he has to maintain the WSN after its deployment. After three months, as his requirement of agricultural field monitoring is completed, he removes the WSN from the agricultural field. Six months later, John plans to use the WSN that was deployed in the agricultural field for home surveillance.

As the agriculture application is different from the home surveillance application, the sensor required for the system also changes. Thus, John has to go through all the steps again, including maintenance, deployment, and hardware management, for the surveillance system. Thus, we observe that the users face different responsibilities for using a WSN for an application. In such a situation, if sensor-cloud architecture is present, John can easily use WSNs for his application on a rental basis. Moreover, through the use of sensor-cloud, John can easily switch the application without any manual intervention. On the other end, service providers of the sensor-cloud infrastructure may serve multiple users with the same sensors and earn profit.

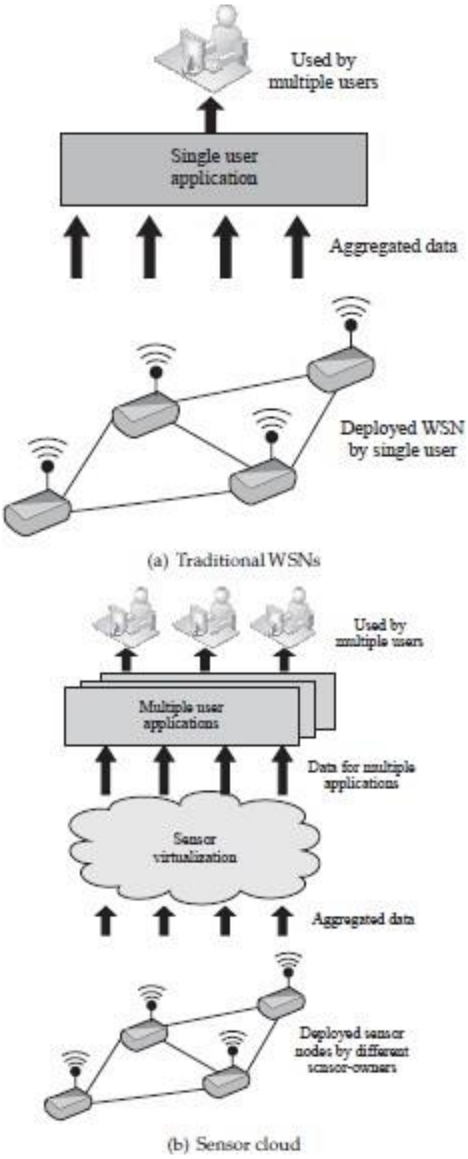


Figure 4.6 Traditional WSN versus sensor-cloud

Architecture of a sensor-cloud platform

In a traditional cloud computing architecture, two actors, cloud service provider (CSP) and end users (customer) play the key role. Unlike cloud computing, in sensor-cloud architecture, the sensor owners play an important role along with the service provider and end users. However, a service provider in sensor-cloud architecture is known as a sensor-cloud service provider (SCSP). The detailed architecture of a sensor-cloud is depicted in Figure 4.7.

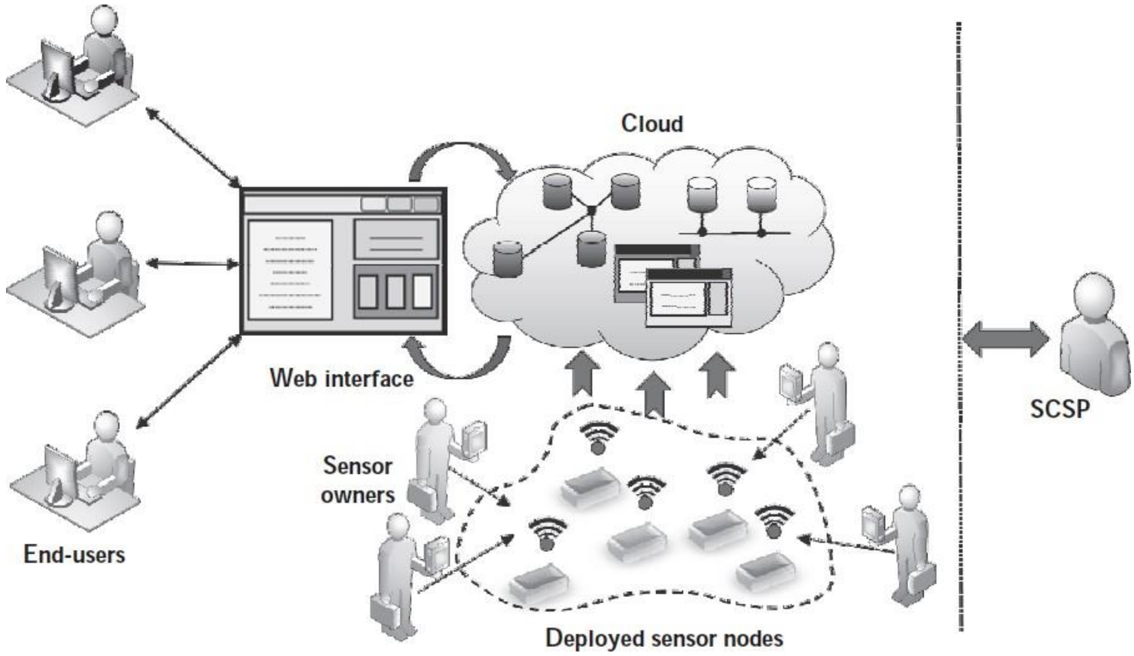


Figure 4.7 Architecture of a sensor-cloud platform

Actors in sensor-cloud architecture

Typically, in a sensor-cloud architecture, three actors are present. We briefly describe the role of each actor.

- **End User:** This actor is also known as a customer of the sensor-cloud services. Typically, an end user registers him/herself with the infrastructure through a Web portal. Thereafter, he/she chooses the template of the services that are available in the sensor-cloud architecture to which he/she is registered. Finally, through the Web portal, the end user receives the services, as shown in Figure 4.7. Based on the type and usage duration of service, the end user pays the charges to the SCSP.
- **Sensor Owner:** We have already discussed that the sensor-cloud architecture is based on the concept of Se-aaS. Therefore, the deployment of the sensors is essential in order to provide services to the end users. These sensors in a sensor cloud architecture are owned and deployed by the sensor owners, as depicted in Figure 4.7. A particular sensor owner can own multiple homogeneous or heterogeneous sensor nodes. Based on the requirements of the users, these sensor nodes are virtualized and assigned to serving multiple applications at the same time. On the other hand, a sensor owner receives rent depending upon the duration and usage of his/her sensor node(s).
- **Sensor-Cloud Service Provider (SCSP):** An SCSP is responsible for managing the entire sensor-cloud infrastructure (including management of sensor owners and end users handling, resource handling, database management, cloud handling etc.), centrally. The SCSP receives rent from end users with the help of a pre-defined pricing model. The pricing scheme may include the infrastructure cost, sensor owners' rent, and the revenue of the SCSP. Typically, different algorithms are used for managing the entire infrastructure. The SCSP receives the rent from the end users and shares a partial amount with the sensor owners. The remaining amount is used for maintaining the infrastructure. In the process, the SCSP earns a certain amount of revenue from the payment of the end users.

Sensor-Cloud Architecture from Different Viewpoints

We explore the sensor-cloud architecture from two view points: (i) User organizational view and (ii) real architectural view [5]. Different views of sensor-cloud architecture are shown in Figure 4.8.

- **User Organizational View:** This view of sensor-cloud architecture is simple. In a sensor-cloud, end users interact with a Web interface for selecting templates of the services. Thereafter, the services are received by the end users through the Web interface. In this architecture, an end user is unaware of the complex processes that are running at the back end.
- **Real Architectural View:** The complex processing of sensor-cloud architecture is visualized through this view. The processes include sensor allocation, data extraction from the sensors, virtualization of sensor nodes, maintenance of the infrastructure, data center management, data caching, and others. For each process, there is a specific algorithm or scheme.

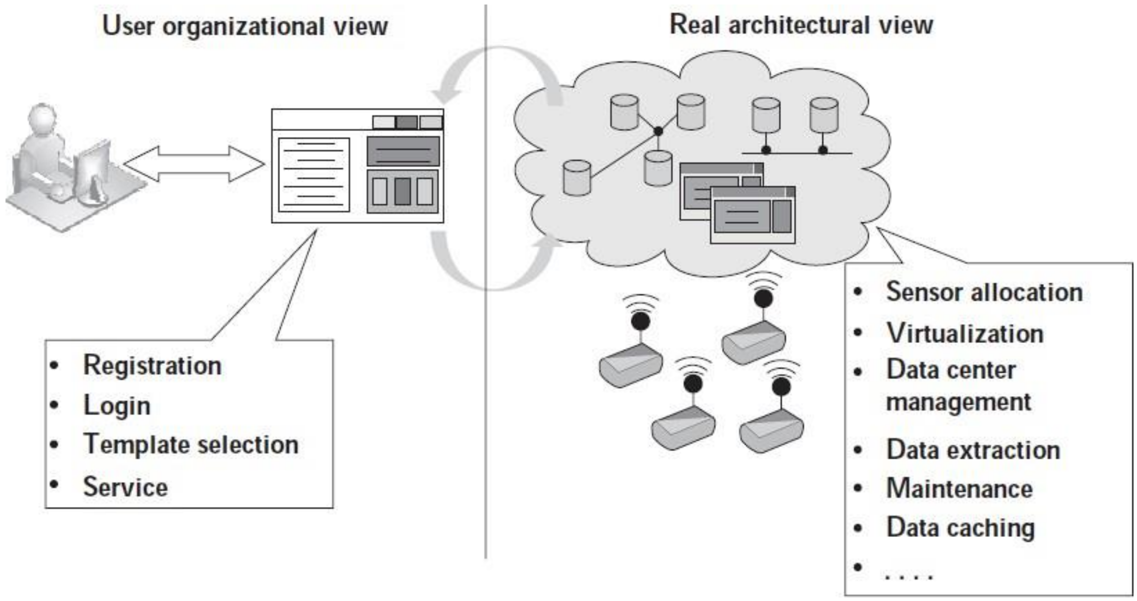


Figure 4.8 Sensor-cloud architecture

➢ IOT CASE STUDIES

Agricultural IoT – Introduction and Case Studies

INTRODUCTION

Currently, IoT-enabled technologies are widely used for increasing crop productivity, generating significant revenue, and efficient farming. The development of the IoT paradigm helps in precision farming. Agricultural loT systems perform crop health monitoring, water management, crop security, farming vehicle tracking, automatic seeding, and automatic pesticide spraying over the agricultural fields. In an IoT based agricultural system, different sensors necessarily have to be deployed over agricultural fields, and the sensed data from these sensors need to be transmitted to a centralized entity such as a server, cloud, or fog devices. Further, these data have to be processed and analyzed to provide various agricultural services. Finally, a user should be able to access these services from handheld devices or computers. Figure 4.9 depicts a basic architecture of an agricultural IoT.

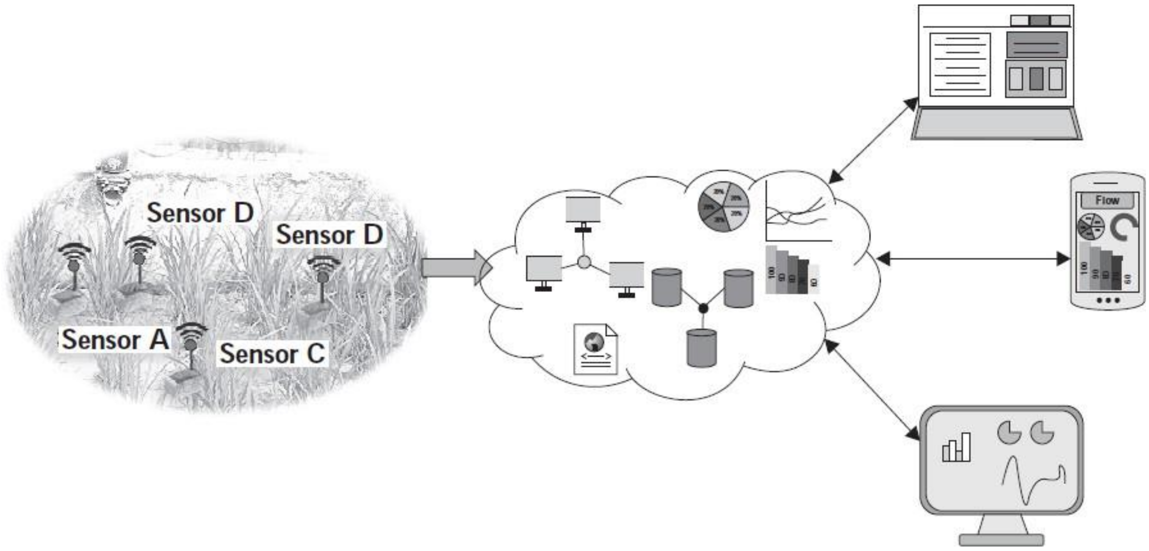


Figure 4.9 Architecture of agricultural IoT

Components of an agricultural IoT

The development of an agricultural IoT has helped farmers enhance crop productivity and reduce the overhead of manual operations of the agricultural equipment in the fields. Different components such as analytics, drone, cloud computing, sensors, hand-held devices, and wireless connectivity enable agricultural IoT as depicted in Figure 4.10.

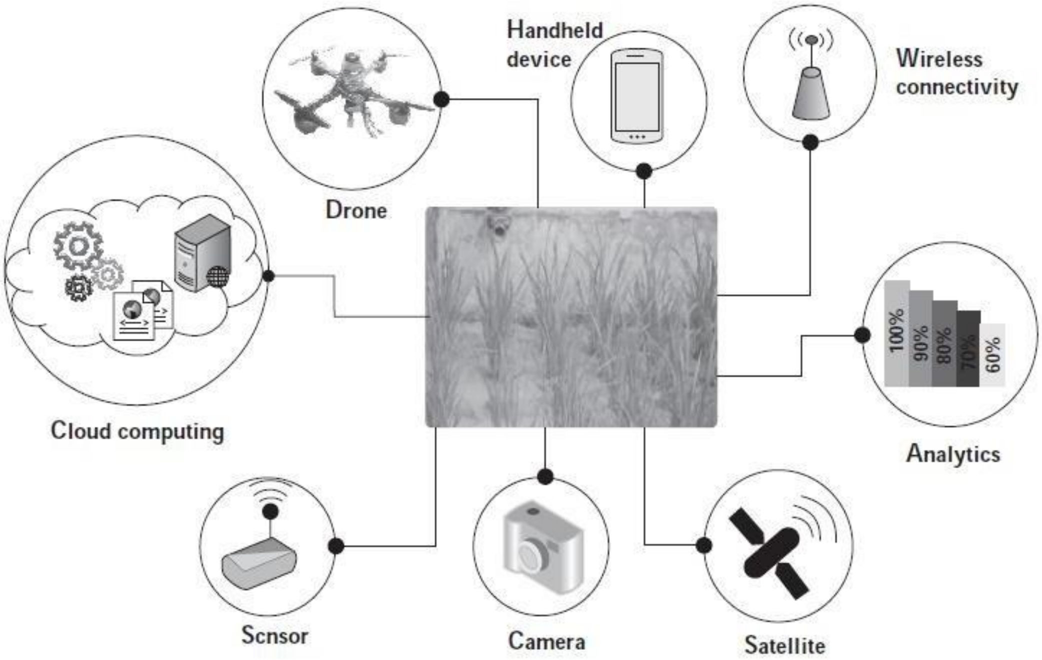


Figure 4.10 Components of agricultural IoT

The different components of an agricultural IoT are discussed as follows:

- **Cloud computing:** Sensors such as the camera, devices to measure soil moisture, soil humidity, and soil pH-level are used for serving different agricultural applications. These sensors produce a huge amount of agricultural data that need to be analyzed. Sometimes, based on the data analysis, action needs to be taken, such as switching on the water pump for irrigation. Further, the data from the deployed sensors are required to be stored on a long-term basis since it may be useful for serving future applications. Thus, for agricultural data analysis and storage, the cloud plays a crucial role.

- **Sensors:** In previous chapters, we already explored different types of sensors and their respective requirements in IoT applications. We have seen that the sensors are the major backbone of any IoT application. Similarly, for agricultural IoT applications, the sensors are an indispensable component. A few of the common sensors used in agriculture are sensors for soil moisture, humidity, water level, and temperature.
- **Cameras:** Imaging is one of the main components of agriculture. Therefore, multispectral, thermal, and RGB cameras are commonly used for scientific agricultural IoT. These cameras are used for estimating the nitrogen status, thermal stress, water stress, and crop damage due to inundation, as well as infestation. Video cameras are used for crop security.
- **Satellites:** In modern precision agriculture, satellites are extensively used to extract information from field imagery. The satellite images are used in agricultural applications to monitor different aspects of the crops such as crop health monitoring and dry zone assessing over a large area.
- **Analytics:** Analytics contribute to modern agriculture massively. Currently, with the help of analytics, farmers can take different agricultural decisions, such as estimating the required amount of fertilizer and water in an agricultural field and estimating the type of crops that need to be cultivated during the upcoming season. Moreover, analytics is not only responsible for making decisions locally; it is used to analyze data for the entire agricultural supply chain. Data analytics can also be used for estimating the crop demand in the market.
- **Wireless connectivity:** One of the main components of agricultural IoT is wireless connectivity. Wireless connectivity enables the transmission of the agricultural sensor data from the field to the cloud/server. It also enables farmers to access various application services over handheld devices, which rely on wireless connectivity for communicating with the cloud/server.
- **Handheld devices:** Over the last few years, e-agriculture has become very popular. One of the fundamental components of e-agriculture is a handheld device such as a smartphone. Farmers can access different agricultural information, such as soil and crop conditions of their fields and market tendency, over their smartphones. Additionally, farmers can also control different field equipment, such as pumps, from their phones.
- **Drones:** Currently, the use of drones has become very attractive in different applications such as surveillance, healthcare, product delivery, photography, and agriculture. Drone imaging is an alternative to satellite imaging in agriculture. In continuation to providing better resolution land mapping visuals, drones are used in agriculture for crop monitoring, pesticide spraying, and irrigation. An agricultural food chain (agri-chain) represents the different stages that are involved in agricultural activity right from the agricultural fields to the consumers.

Figure 4.11 depicts a typical agricultural food chain with the different operations that are involved in it. Additionally, the figure depicts the applications of different IoT components required for performing these agricultural operations. In the agrichain,we consider farming as the first stage. In farming, various operations, such as seeding, irrigation, fertilizer spreading, and pesticide spraying, are involved. For performing these operations, different IoT components are used. As an example, for monitoring the soil health, soil moisture and temperature sensors are used; drones are used for spraying pesticides; and through wireless connectivity, a report on on-field soil conditions is sent directly to a users' handheld device or cloud. After farming, the next stage in the agri-chain is transport. Transport indicates the transfer of crops from the field to the local storage, and after that, to long-term storage locations. In transport, smart vehicles can automatically load and unload crops. The global positioning system (GPS) plays an important role by tracking these smart devices, and radio frequency identification (RFID) is used to collect information regarding the presence of a particular container of a crop at a warehouse.

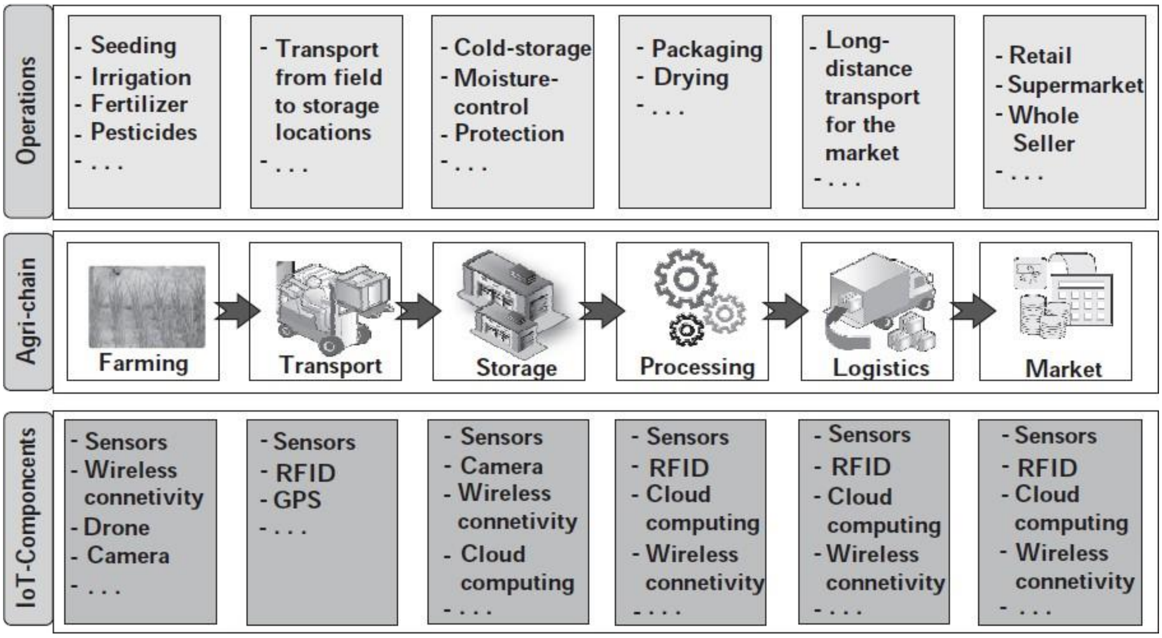


Figure 4.11 Use of IoT components in the agricultural chain

Storage is one of the important operations in the agri-chain. It is responsible for storing crops on a long term basis. Typically, cold storage is used for preserving the crops for a long time and providing them with the necessary climatic and storage conditions and protection. In the storage, cameras are used to keep a check and protect the harvested crops. The camera feeds are transferred through wireless connectivity to a remote server or a cloud infrastructure. Moreover, the amount and type of crops stored in a storage location are tracked and recorded with the help of sensors and cloud computing. For pushing the crops into the market, processing plays a crucial role in an agrichain. Processing includes proper drying and packaging of crops. For drying and packaging, different sensors are used. Packaging is the immediate operation prior to pushing the crop into the market. Thus, it is essential to track every package and store all the details related to the crops in the cloud. Logistics enables the transfer of the packed crops to the market with the help of smart vehicles. These smart vehicles are equipped with different sensors that help in loading and unloading the packed crop autonomously. Additionally, GPS is used in these smart vehicles for locating the position of the packed crops at any instant and tracking their whereabouts. All the logistical information gets logged in the cloud with the help of wireless connectivity. Finally, the packed items reach the market using logistical channels. From the market, these items are accessible to consumers. The details of the sale and purchase of the items are stored in the form of records in the cloud.

Advantages of IoT in agriculture

Modern technological advancements and the rapid developments in IoT components have gradually increased agricultural productivity. Agricultural IoT enables the autonomous execution of different agricultural operations. The specific advantages of the agricultural IoT are as follows:

- **Automatic seeding:** IoT-based agricultural systems are capable of autonomous seeding and planting over the agricultural fields. These systems significantly reduce manual effort, error probability, and delays in seeding and planting.
- **Efficient fertilizer and pesticide distribution:** Agricultural IoT has been used to develop solutions that are capable of applying and controlling the amount of fertilizers and pesticides efficiently. These solutions are based on the analysis of crop health.
- **Water management:** The excess distribution of water in the agricultural fields may affect the growth of crops. On the other hand, the availability of global water resources is finite. The constraint of limited and often scarce usable water resources is an influential driving factor for the judicious and efficient distribution of agricultural water resources. Using the various solutions available for agricultural IoT, water can be distributed efficiently, all the while, increasing field productivity and yields. The IoT-enabled agricultural systems are capable of monitoring the water level and moisture in the soil, and accordingly, distribute the water to the agricultural fields.
- **Real-time and remote monitoring:** Unlike traditional agriculture, in IoT-based farming, a stakeholder can remotely monitor different agricultural parameters, such as crop and soil conditions, plant health, and weather conditions. Moreover, using a smart handheld device (e.g., cellphone), a farmer can actuate on-field farming machinery such as a water pump, valves, and other pieces of machinery.
- **Easy yield estimation:** Agricultural IoT solutions can be used to record and aggregate data, which may be spatially or temporally diverse, over long periods. These records can be used to come up with various estimates related to farming and farm management. The most prominent among these estimates is crop yield, which is done based on established crop models and historical trends.
- **Production overview:** The detailed analysis of crop production, market rates, and market demand are essential factors for a farmer to estimate optimized crop yields and decide upon the essential steps for future cropping practices. Unlike traditional practices, IoT-based agriculture acts as a force multiplier for farmers by enabling them to have a stronger hold on their farming as well as crop management practices, and that too mostly autonomously. Agricultural IoT provides a detailed product overview on the farmers' handheld devices.

CASE STUDIES

In this section, we discuss a few case studies that will provide an overview of real implementation of IoT infrastructure for agriculture.

In-situ assessment of leaf area index using IoT-based agricultural system

In this case study, we focus on an IoT-based agricultural system developed by Bauer. The authors focus on the in-situ assessment of the leaf area index (LAI), which is considered as an essential parameter for the growth of most crops. LAI is a dimensionless quantity which indicates the total leaf area per unit ground area. For determining the canopy (the portion of the plant, which is above the ground) light, LAI plays an essential role.

Architecture

The authors integrated the hardware and software components of their implementation in order to develop the IoT-based agricultural system for LAI assessment. One of the important components in this system is the wireless sensor network (WSN), which is used as the LAI assessment unit. The authors used two types of sensors: (i) ground-level sensor (G) and (ii) reference sensor (R).

These sensors are used to measure photosynthetically active radiation (PAR). The distance between the two types of sensors must be optimal so that these are not located very far from one another. In this system, the above-ground sensor (R) acts as a cluster head while the other sensor nodes (Gs) are located below the canopy. These Gs and R connect and form a star topology. A solar panel is used to charge the cluster head. The system is based on IoT architecture. Therefore, a cluster head is attached to a central base station, which acts as a gateway. Further, this gateway connects to an IoT infrastructure. The architecture of the system is depicted in Figure 4.12.

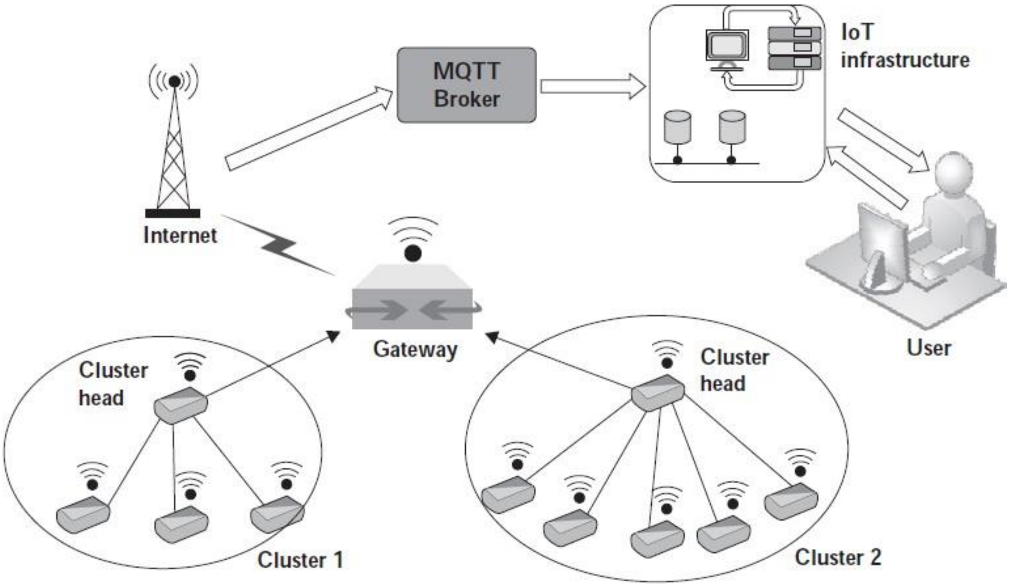


Figure 4.12 System architecture

Hardware

For sensing and transmitting the data from the deployment fields to a centralized unit, such as a server and a cloud, different hardware components are used in the system. The commercial off-the-shelf (COTS) TelosB platform is used in the system. The TelosB motes are equipped with three types of sensors: temperature, humidity, and light sensors. With the help of an optical filter and diffuser accessory on the light sensors, the PAR is calculated to estimate the LAI. The system is based on the cluster concept. A Raspberry-Pi is used as a cluster head, which connects with four ground sensor motes. The Raspberry-Pi is a tiny single board, which works as a computer and is used to perform different operations in IoT. Humidity and wet plants intermittently cause attenuation to the system, which is minimized with the help of forward error coding (FEC) technique. The real deployment of the LAI assessment system involves various environmental and wild-life challenges. Therefore, for reliable data delivery, the authors take the redundant approach of using both wired and wireless connectivity. In the first deployment generation, USB power supply is used to power-up the sensors motes. Additionally, the USB is used for configuring the sensor board and accessing the failure as per requirement. In this setup, a mechanical timer is used to switch off the sensor nodes during the night. In the second deployment generation, the cluster is formed with wireless connectivity. The ground sensor motes consist of external antennas, which help to communicate with the cluster head. A Raspberry-Pi with long-term evolution (LTE) is used as a gateway in this system.

Communication

The LAI system consists of multiple components, such as WSN, IoT gateway, and IoT based network. All of these components are connected through wired or wireless links. The public land mobile network (PLMN) is used to establish connectivity between external IoT networks and the gateway. The data are analyzed and visualized with the help of a farm management information system (FMIS), which resides in the IoT-based infrastructure. Further, a prevalent data transport protocol: MQTT, is used in the system. MQTT is a very light-weight, publish/subscribe messaging protocol, which is widely used for different IoT applications. The wireless LAN is used for connecting the cluster head with a gateway. The TelosB motes are based on the IEEE 802.15.4 wireless protocol.

Software

Software is an essential part of the system by which different operations of the system are executed. In order to operate the TelosB motes, TinyOS, an open-source, low-power operating system, is used. This OS is widely used for different WSN applications. Typically, in this system, the data acquired from the sensor node is stored with a timestamp and sequence number (SN). For wired deployments (the first generation deployment), the sampling rate used is 30 samples/hour. However, in the wireless deployment (the second generation), the sampling rate is significantly reduced to 6 samples/hour. The TinyOS is capable of activating low-power listening modes of a mote, which is used for switching a mote into low-power mode during its idle state. In the ground sensor, TelosB motes broadcast the data frame, and the cluster head (Raspberry-Pi) receives it. This received data is transmitted to the gateway. Besides acquiring ground sensor data, the Raspberry-Pi works as a cluster head. In this system, the cluster head can re-boot any affected ground sensor node automatically.

IoT Architecture

The MQTT broker runs in the Internet server of the system. This broker is responsible for receiving the data from the WSN. In the system, the graphical user interface (GUI) is built using an Apache server. The visualization of the data is performed at the server itself. Further, when a sensor fails, the server informs the users. The server can provide different system-related information to the smartphone of the registered user.

Smart irrigation management system

In precision agriculture, the regular monitoring of different agricultural parameters, such as water level, soil moisture, fertilizers, and soil temperature are essential. Moreover, for monitoring these agricultural parameters, a farmer needs to go to his/her field and collect the data. Excess water supply in the agricultural field can damage the crops. On the other hand, insufficient water supply in the agricultural field also affects the healthy growth of crops. Thus, efficient and optimized water supply in the agricultural field is essential. This case study highlights a prototype of an irrigation management system developed at the Indian Institute of Technology Kharagpur, funded by the Government of India. The primary objective of this system is to provide a Web-based
platform to the farmer for managing the water supply of an irrigated agricultural field. The system is capable of providing a farmer-friendly interface by which the field condition can be monitored. With the help of this system, a farmer can take the necessary decision for the agricultural field based on the analysis of the data. However, the farmer need not worry about the complex background architecture of the system. It is an affordable solution for the farmers to access the agricultural field data easily and remotely.

Architecture

The architecture of this system consists of three layers: Sensing and actuating layer, remote processing and service layer, and application layer. These layers perform dedicated tasks depending on the requirements of the system. Figure 4.13 depicts the architecture of the system. The detailed functionalities of different layers of this system are as follows:

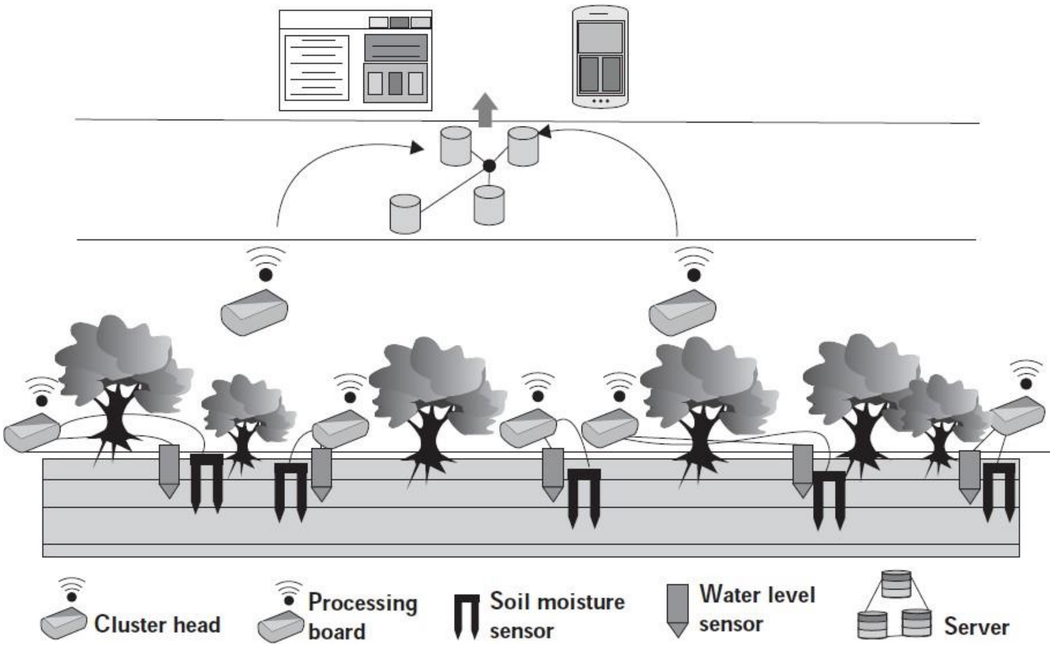
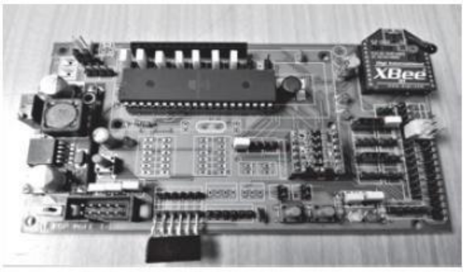


Figure 4.13 Architecture: Smart irrigation management system

Sensing and Actuating layer: This layer deals with different physical devices, such as sensor nodes, actuators, and communication modules. In the system, a specially designated sensor node works as a cluster head to collect data from other sensor nodes, which are deployed on the field for sensing the value of soil moisture and water level. A cluster head is equipped with two communication module: ZigBee (IEEE 802.15.4) and General Packet Radio Service (GPRS). The communication between the deployed sensor nodes and the cluster head takes place with the help of ZigBee. Further, the cluster heads use GPRS to transmit data to the remote server. An electrically erasable programmable read-only memory (EEPROM), integrated with the cluster head, stores a predefined threshold value of water levels and soil moisture. When the sensed value of the deployed sensor node drops below this predefined threshold value, a solenoid (pump) activates to start the irrigation process. In the system, the standard EC-05 soil moisture sensor is used along with the water level sensor, which is specifically designed and developed for this project. A water level sensor is shown in Figure 4.14(a).

Processing and Service layer: This layer acts as an intermediate layer between the sensing and actuating layer and the application layer. The sensed and process data is stored in the server for future use. Moreover, these data are accessible at any time from any remote location by authorized users. Depending on the sensed values from the deployed sensor nodes, the pump actuates to irrigate the field. A processing board as depicted in Figure 4.14(b) is developed for the project.

(a) Water level sensor



(b) Processing board

Figure 4.14 Water level sensor and processing board

Application layer: The farmer can access the status of the pump, whether it is in switch on/off, and the value of different soil parameters from his/her cell phone. This information is accessible with the help of the integrated GSM facility of the farmers' cell phone. Additionally, an LED array indicator and LCD system is installed in the farmers' house. Using the LCD and LED, a farmer can easily track the condition of his respective fields. Apart from this mechanism, a farmer can manually access field information with the help of a Web-based application. Moreover, the farmer can control the pump using his/her cell phone from a remote location.

Deployment

The system has been deployed and experimented in two agricultural fields: (i) an agricultural field at the Indian Institute of Technology Kharagpur (IIT Kharagpur), India, and (ii) Benapur, a village near IIT Kharagpur, India. Both the agricultural fields were divided into 10 equal sub-fields of $3x3m^2$. In order to examine the performance, the system was deployed at over 4 sub-fields. Each of these sub-fields consists of a solenoid valve, a water level sensor, and a soil moisture sensor, along with a processing board. On the other hand, the remaining six sub-fields were irrigated through a manual conventional irrigation process. The comparison analysis between these six and four fields summarily reports that the designed system's performance is superior to the conventional manual process of irrigation.