K - 52362

Register No. | | | | | | | | |

BE Degree Examination May 2023

Sixth Semester

Computer Science and Engineering

20CST62 – INTERNET OF THINGS AND CLOUD

(Regulations 2020)

Time: Three hours

Maximum: 100 marks

Answer all Questions

Part – A (10 × 2 = 20 marks)

1. Define IoT. [CO1,K1]
2. Compare IoT and M2M. [CO1,K2]
3. How do Bluetooth beacons work? [CO2,K2]
4. Why service discovery is very important in IoT? [CO2,K2]
5. What is thingspeak and how it works? [CO3,K2]
6. Write a python code to blink led 10 times on button click. [CO3,K3]
7. Why is hybrid cloud flexible? [CO4,K2]
8. Specify the benefits of AWS IoT analytics. [CO4,K2]
9. What is the purpose of device shadow? [CO5,K1]
10. Name any four cloud service provider. [CO5,K1]

Part – B (5 × 16 = 80 marks)

11. a. Consider an IoT based smart irrigation monitoring and control system that monitors and controls the supply of water from a remote location. For this IoT, apply the following design methodologies (16) [CO1,K3]

1) Purpose and requirement specification

2) Process specification

3) Domain mode specification

4) IoT level specification

(OR)

b. An IoT based ICU patient monitoring system that collects patients information with the help of sensons. The system needs to monitor various parameters like blood pressure, heart, pulse rate and temperature of the patient. The monitored parameters are sent to the cloud using internet so that the doctors and patients can view the details from anywhere else. For this scenario, suggest a suitable IoT level and provide the justification of chosen level. (16) [CO1,K3]

12.	a.	i)	Explain briefly about layered architecture of IoT.	(10)	[CO2,K2]
		ii)	Sketch the categorization of IoT protocols.	(6)	[CO2,K2]
			(OR)		
	b.	i)	Illustrate the network architecture of 6LOWAAN with neat sketch.	(8)	[CO2,K2]
		ii)	Explain the protocols that are prominently used in IoT service discovery.	(8)	[CO2,K2]
13.	a.		Suppose you want to automate the traffic light system at junction where there is a huge traffic. Develop a python program that simulates the traffic controller system using Raspberry Pi and upload the status to thingspeak cloud.	(16)	[CO3,K3]
			(OR)		
	b.		Interface a light sensor with Raspberry Pi that detects the light intensity of a room. If the intensity goes below a threshold, the intensity has to be uploaded to the thingspeak cloud. Develop a python program for this.	(16)	[CO3,K3]
14.	a.	i)	Investigate the architectural style for implementing cloud federation.	(6)	[CO4,K2]
		ii)	Illustrate the various architectural components of the smarter traffic system and explain each component in detail.	(10)	[CO4,K2]
			(OR)		
	b.		Discuss in detail about cloud inspired IoT solutions available for smarter environments.	(16)	[CO4,K2]
15.	a.		Explain the steps involved in creation of the following 1) AWS IoT - Policy 2) AWS IoT - Thing Object.	(16)	[CO5,K2]
			(OR)		
	b.		Briefly explain the interfaces available for devices and apps to access AWS IoT.	(16)	[CO5,K2]

Bloom's Taxonomy Level	Remembering (K1)	Understanding (K2)	Applying (K3)	Analysing (K4)	Evaluating (K5)	Creating (K6)
Percentage	3	60	37	-	-	-

B.E DEGREE EXAMINATION MAY 2023
Sixth Semester
Computer Science and Engineering
20CST62 Internet of Things and Cloud
Regulation 2020

PART-A

ANSWER ALL THE QUESTIONS

1. Define IoT

A dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual things have identities, physical attributes and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network, often communicate data associated with users and their environments.

2. Compare IoT and M2M (any 2 points)

CRITERIA	M2M	IoT
Communication	Focus below network layer	Focus above network layer
Machines	Homogeneous machine types	Heterogeneous machine types
H/W Vs S/W	Emphasis of M2M is more on hardware	IoT is more on software
Storage	Point solutions and often in on-premises storage	Data in IoT is collected in the cloud

3. How do Bluetooth beacons work?

A beacon device sends out a signal consisting of a unique identifier, which belongs to that particular beacon. No response or signal is sent from the user's device back to the beacon.

4. Why service discovery is very important in IoT?

- Service Discovery is a process of automatically finding appropriate services and their providers
- In IoT, many heterogeneous objects offer different services so it is challenging to locate desirable services due to the considerable diversity and large scale

5. What is thingspeak and how its work?

- ThingSpeak is an IoT analytics platform service
- It allows to aggregate, visualize and analyze live data streams in the clou
- Channel: Private or Public
- API keys

6. Write a python code to blink led 10 times on button click.

```
import RPi.GPIO as GPIO
from time import sleep
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(8, GPIO.OUT, initial=GPIO.LOW)
for i in range(10):
    GPIO.output(8, GPIO.HIGH)
    sleep(1)
    GPIO.output(8, GPIO.LOW)
  sleep(1)
```

7. Why is hybrid cloud flexible?

- Hybrid cloud combination of both private and public cloud
- Sensitive information can be stored Private and Non-Sensitive information can be stored Public
-So, hybrid cloud is flexible

8. Specify the benefits of AWS IoT analytics

- Better visibility and control result in faster decision-making.
- Scalability of business requirements and expansion into other markets.
- Automation results in lower operational costs and greater resource utilization.
- New revenue streams as a result of operational difficulties being resolved.

9. What is the purpose of device shadow?

- Device Shadows can make a device's state available to apps and other services whether the device is connected to AWS IoT or not
- Synchronisation of devices

10. **Name any four cloud service providers**

Microsoft Azure. ...
Google Cloud Platform. ...
Alibaba Cloud. ...
Salesforce. ...
IBM. ...
Digital Ocean. .

PART-B (5 x 16 = 80 marks)

11.a. Consider an IoT based smart irrigation monitoring and control system that monitors and controls the supply of water from a remote location. For this IoT, apply the following design methodologies. (16 Marks)

Design Methodologies for IoT-based Smart Irrigation Monitoring and Control System:

1) Purpose and Requirement Specification: (4 Marks)

Purpose: To monitor and control the supply of water for irrigation remotely. (2Marks)

Requirements: (2Marks)

- Real-time monitoring of soil moisture levels
- Automatic control of water supply based on soil moisture data
- Remote access and control through a mobile or web application.
- Alerts and notifications for critical events (e.g., low water level, system malfunction).
- Data logging and analytics for historical analysis and decision-making.
- Energy-efficient operation to conserve power.
- Scalability to support multiple irrigation zones or areas.

2) Process Specification: (4 Marks)

- Use cases of the IoT system are formally described based on and derived from the purpose and requirement specifications (2 Marks)
- Diagram (2 Marks)

3) Domain Model Specification: (4 Marks)

- Describes the **main concepts**, **entities** and **objects** in the domain of IoT system to be designed
- Defines the **attributes** of the objects and **relationships** between objects
- Provides an **abstract representation** of the concepts, objects and entities in the IoT domain, independent of any specific technology or platform (2 Marks)
- Diagram (2 Marks)

4) IoT Level Specification: (4 Marks)

- **Level Identification- 4/5/6** (2 Marks)
- Justification (2 Marks)

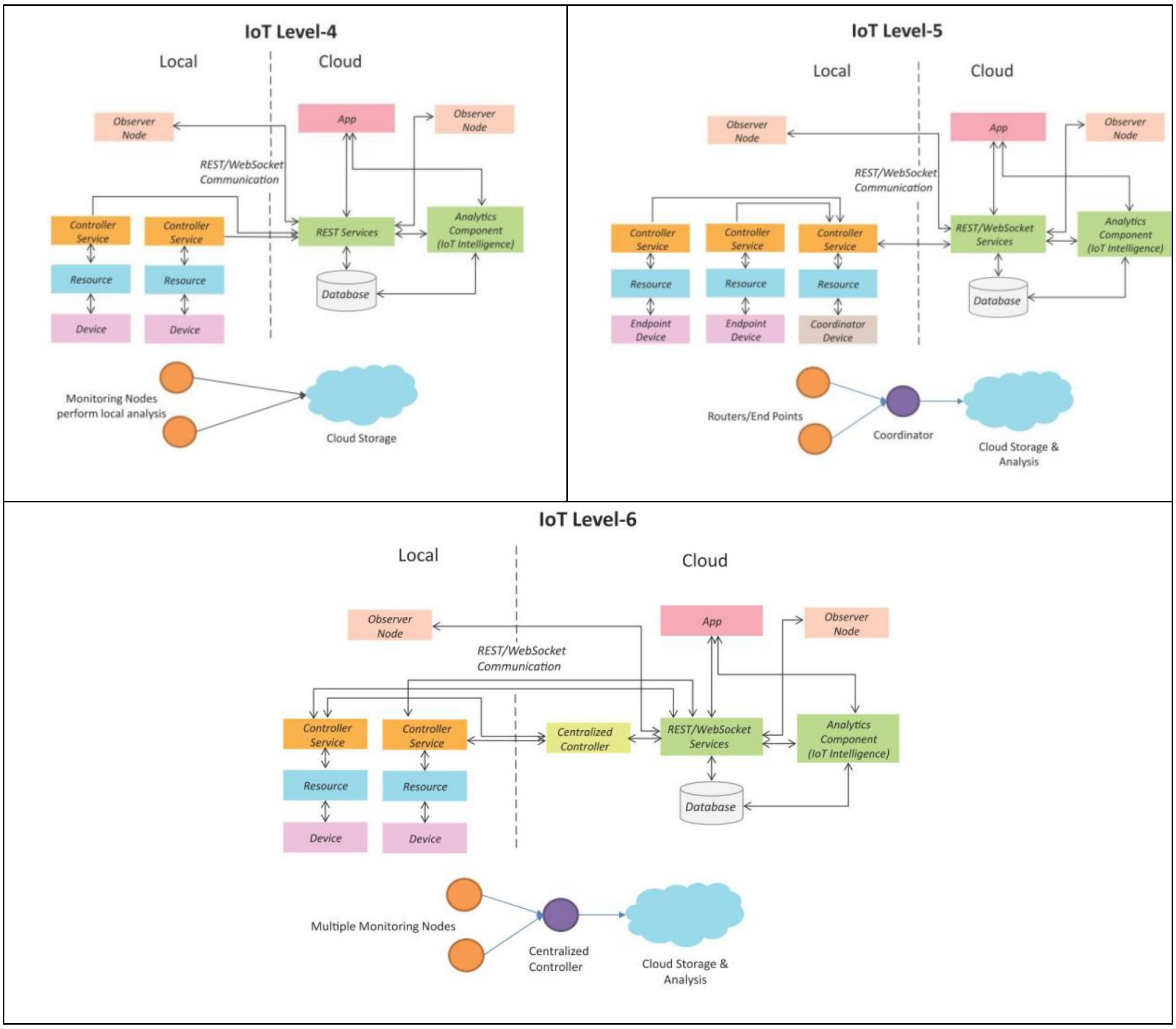
OR

11.b. An IOT based ICU patient monitoring system that collects patient's information with the help of sensors. The system needs to monitor various parameters like blood pressure, heart, pulse rate and temperature of the patient. The monitored parameters

are sent to the cloud using Internet so that the doctors and patients can view the details from anywhere else. For this scenario, suggest a suitable IoT level and provide the justification of chosen model.

Level Identification -4/5/6 (2 Marks)

Diagram for identified level: (8 Marks)

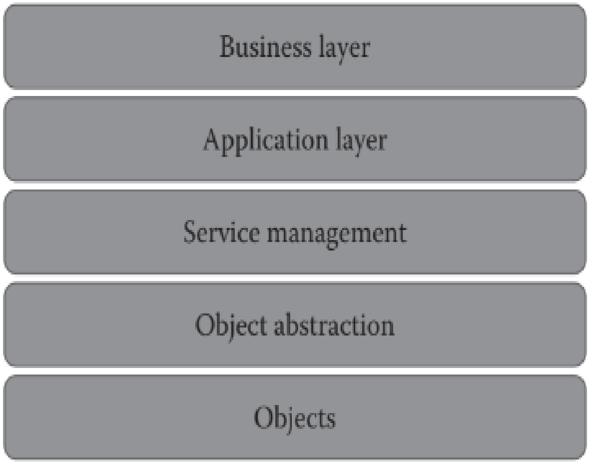


Justification- 6 Marks

- Multiple nodes (Blood pressure, heart pulse rate, temperature sensors)
- Cloud (Storage/ storage and analysis)
- Observer Node
- Co-ordinator / centralised co-ordinator

12.a.i. Explain briefly about layered architecture of IoT. (10 Marks)

Diagram- 3 Marks



Explanation – 7 Marks

Objects Layer (2 Marks**)**

- Objects layer, also known as devices layer
- It comprises the physical devices that are used to collect and process information from the IoT ecosystem

Object Abstraction Layer (2 Marks**)**

- This layer transfers data that are collected from objects to service management layer using secure transmission channels.
  Data transmission can happen using any of the following technologies:
  - RFID
  - 3G
  - GSM
  - UMTS
  - Wi-Fi
  - Bluetooth low energy
  - Infrared
  - ZigBee

Service Management Layer (1 Mark**)**

Middleware for the IoT ecosystem

Application layer (1 Mark**)**

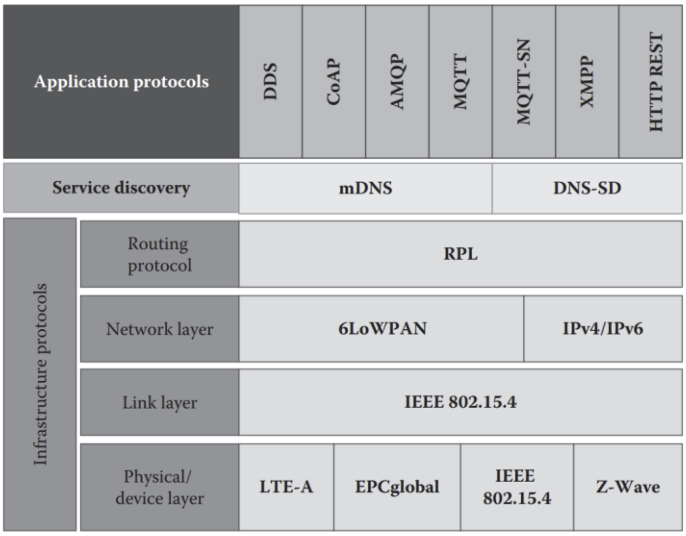
Provides the diverse kinds of services requested by the customer.

Business Layer (1 Mark**)**

This layer uses the data that are received from the network layer to build various components such as business models, graphs, and flowcharts.

12.a.ii. Sketch the categorization of IoT protocols (6 Marks)

Diagram- 4 Marks



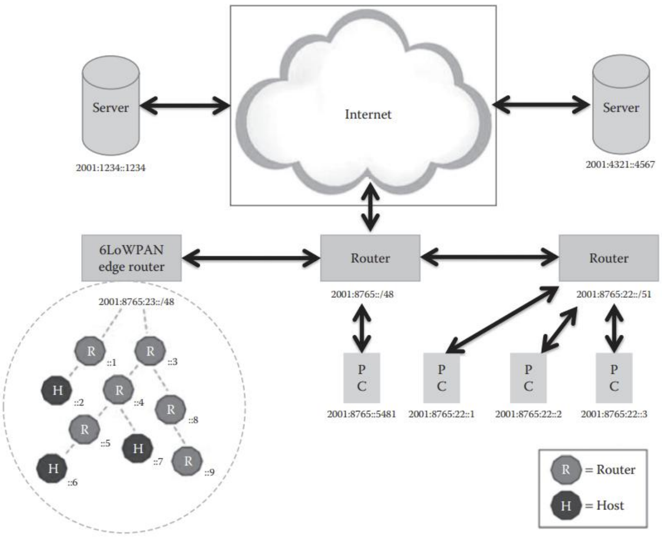
Explanation about IoT specific protocols - 2 Marks

MQTT – Message Queue Telemetry Transport Protocol
DDS – Data Distribution Service
AMQP – Advanced Message Queuing Protocol
CoAP – Constrained Application Protocol

OR

12. b. i. Illustrate the network architecture of 6LoWPAN with a neat sketch. (8 Marks)

Diagram- 4 Marks



Explanation- 4 Marks

- Devices (Routers and Hosts/ end point devices)
- Functions of edge routers and Hosts

12. b. ii. Explain the protocols that are prominently used in IoT service discovery. (8 Marks)

Service Discovery layer has a prominent role in an IOT architecture. It is the service discovery or service management layer which differentiates an IOT network with that of typical internet network. The IOT devices need to connect and communicate with web or cloud based services and applications for IOT implementation.

Some of the popular service discovery protocols are as follow –

1) mDNS (2 marks)
   mDNS – Multicast Domain Name System (mDNS) is a DNS like service discovery protocol to resolve host names to IP addresses in a local network without using any unicast DNS server. It can be used without any additional infrastructure or DNS server in the network.
2) DNS Service Discovery (DNS-SD) (2 marks)
   This protocol stack uses standard DNS messages to discover services in an IOT network. Based on mDNS, DNS-SD is used to resolve services available in a network. The service discovery is implemented in two steps – in the first step, host names of the service providers are resolved and in the next step, IP addresses are paired with the host names using mDNS. It is important to identify host names as IP addresses can change in the network.
3) uPnP (2 marks)
   Maintained by Open Connectivity Foundation, Universal Plug and Play (uPnP) is a protocol stack that allows devices in a network to discover each other and each other's capabilities along with setting up network functions like data sharing and communication.
4) Simple Discovery Service Protocol (2 marks)
   Simple Discovery Service Protocol is used in UPnP networks for discovering available services. It is used by control points in UPnP network to look for devices, services offered by them and their availability at a time. The control point sends a multicast search request to which the device offering requested service responds. The devices and services offered by them are identified by device and service descriptors respectively. The control point uses devices and service descriptions to look for requested services in the network.

13. a. Suppose you want to automate the traffic light system at junction where there is a huge traffic. Develop the python program that simulates the traffic controller system using Raspberry Pi and upload the status to Thingspeak cloud. (16 Marks)

Traffic light code (alone) – 8 Mark

Cloud integration in the code- 5 Marks
Explanation- 3 Marks

Program:

```
import RPi.GPIO as GPIO
import time
import requests

# Define the GPIO pins for the traffic lights
RED_PIN = 17
YELLOW_PIN = 27
GREEN_PIN = 22

# Define the Thingspeak API endpoint and API key
THINGSPEAK_API_ENDPOINT = "https://api.thingspeak.com/update"
THINGSPEAK_API_KEY = "YOUR_THINGSPEAK_API_KEY"

# Set up GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(RED_PIN, GPIO.OUT)
GPIO.setup(YELLOW_PIN, GPIO.OUT)
GPIO.setup(GREEN_PIN, GPIO.OUT)

# Function to control the traffic lights
def set_traffic_lights(red, yellow, green):
    GPIO.output(RED_PIN, red)
    GPIO.output(YELLOW_PIN, yellow)
    GPIO.output(GREEN_PIN, green)

# Function to upload the traffic light status to Thingspeak
def upload_to_thingspeak(status):
    params = {
        "api_key": THINGSPEAK_API_KEY,
        "field1": status
    }
    requests.post(THINGSPEAK_API_ENDPOINT, params=params)

# Main program loop
while True:
    # Red light
    set_traffic_lights(True, False, False)
    upload_to_thingspeak(0)  # Upload status 0 for red light
    time.sleep(5)  # Stay red for 5 seconds

    # Red and yellow lights
```

```
    set_traffic_lights(True, True, False)
    upload_to_thingspeak(1)  # Upload status 1 for red and yellow lights
    time.sleep(2)  # Stay red and yellow for 2 seconds

    # Green light
    set_traffic_lights(False, False, True)
    upload_to_thingspeak(2)  # Upload status 2 for green light
    time.sleep(5)  # Stay green for 5 seconds

    # Yellow light
    set_traffic_lights(False, True, False)
    upload_to_thingspeak(3)  # Upload status 3 for yellow light
    time.sleep(2)  # Stay yellow for 2 seconds

# Clean up GPIO
GPIO.cleanup()
```

Or

13. b. Interface a light sensor with Raspberry Pi that detects the light intensity of a room, If the intensity goes below a threshold, the intensity has to be uploaded to the thingspeak cloud. Develop a python program for this. (16 Marks)

Steps: 6 Marks

Define the GPIO pin for the LDR
Define the threshold for light intensity
Define the Thingspeak API endpoint and API key
Function to read the light intensity from the LDR
Read the light intensity from the LDR
Check intensity with Threshold
Function to upload the light intensity to Thingspeak

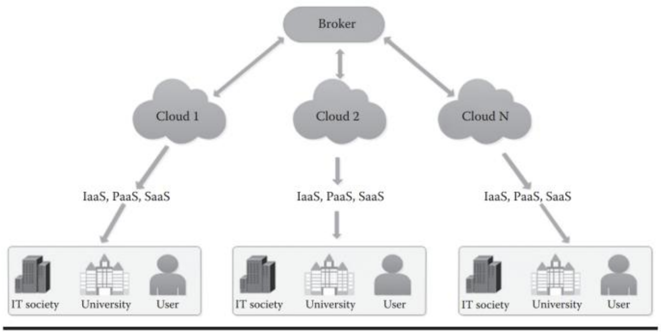
Program: 10 Marks

```
import RPi.GPIO as GPIO
import time
import requests

# Define the GPIO pin for the LDR
LDR_PIN = 18

# Define the threshold for light intensity
INTENSITY_THRESHOLD = 100

# Define the Thingspeak API endpoint and API key
```

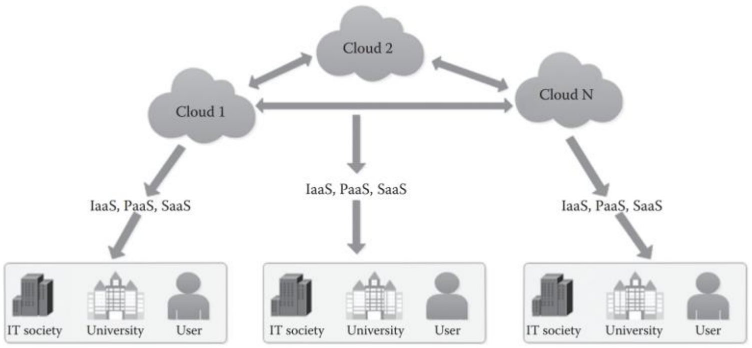
```
THINGSPEAK_API_ENDPOINT = "https://api.thingspeak.com/update"
THINGSPEAK_API_KEY = "YOUR_THINGSPEAK_API_KEY"

# Set up GPIO
GPIO.setmode(GPIO.BCM)

# Function to read the light intensity from the LDR
def read_light_intensity():
    # Set the LDR pin as input
    GPIO.setup(LDR_PIN, GPIO.IN)
    intensity = 0

    # Measure the LDR voltage for a short period of time
    for _ in range(10):
        intensity += GPIO.input(LDR_PIN)
        time.sleep(0.01)

    # Calculate the average intensity
    intensity /= 10

    return intensity

# Function to upload the light intensity to Thingspeak
def upload_to_thingspeak(intensity):
    params = {
        "api_key": THINGSPEAK_API_KEY,
        "field1": intensity
    }
    requests.post(THINGSPEAK_API_ENDPOINT, params=params)

# Main program loop
while True:
    light_intensity = read_light_intensity()

    if light_intensity < INTENSITY_THRESHOLD:
        upload_to_thingspeak(light_intensity)

    time.sleep(10)  # Check the light intensity every 10 seconds

# Clean up GPIO
GPIO.cleanup()
```

14. a. i. Investigate the architectural style for implementing cloud federation (6 Marks)

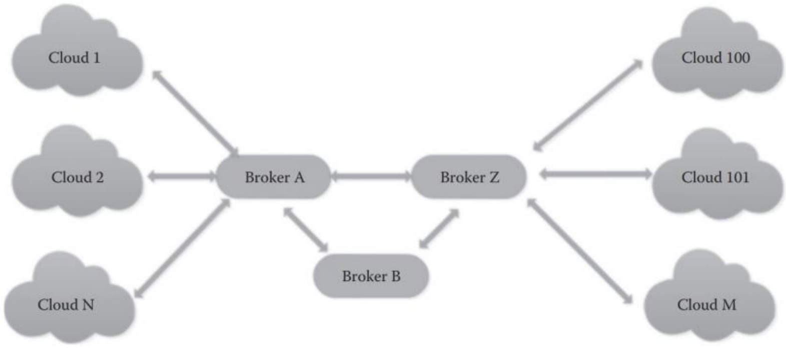
Centralized scheme: Cloud brokers find to bind and create a cloud federation out of multiple clouds to fulfill the requirement. **(2 Marks)**



Decentralized scheme: clouds, in a peer-to-peer manner, negotiate them selves to establish the required partnership to have a cloud federation. **(2 Marks)**



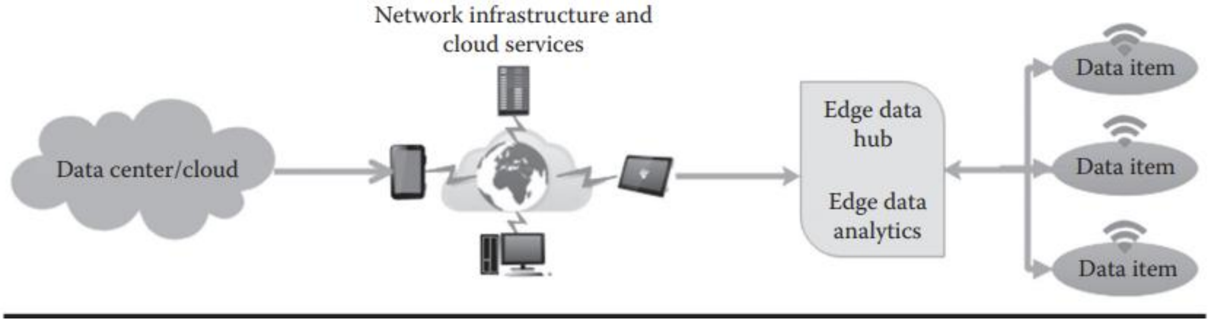
Hierarchical Approach: (2 Marks)



Clouds are connected to a broker and each broker can also interact with other brokers in order to look for fitting clouds for realizing the cloud federation.

14. a. ii. Illustrate the various architectural components of the smarter traffic system and explain each component in detail. (8 Marks)

Diagram-4 Marks

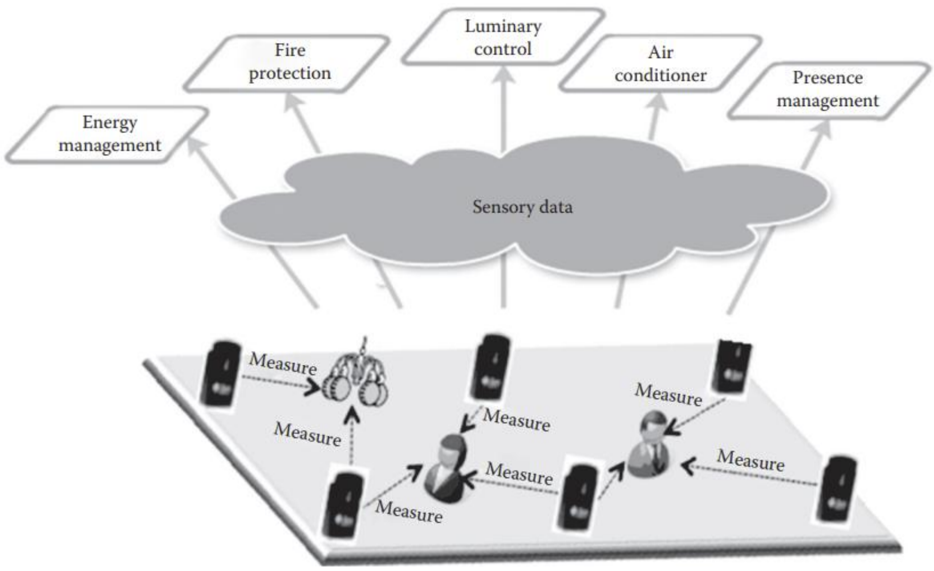


Explanation-6 Marks

1. A smarter traffic system has to include traffic lights, sensors, and actuators within its jurisdictional region so that the reaction time is on the order of <10 milliseconds.
2. M**iniaturized orchestration -** To orchestrate and manage all the other software modules of the system effectively. It has to be policy-aware. That is, well-intended policies can be established easily and enforced accordingly toward effective governance.
3. A **centralized decision-enabling module -** Push the decisions to individual traffic lights through a messaging bus
4. The deviations and deficiencies from multiple smarter traffic systems need to be addressed systematically in order to accomplish edge analytics

OR

14. b. Discuss in detail about cloud inspired IoT solutions available for smarter environments. (16 Marks)



(4 marks)

(Explanation 12)

Smarter Environments

Above figure illustrates how different systems, sensors, and actuators are being quickly integrated to do a real-time capture of disparate and distribute data and leverage them cognitively to arrive at appropriate and actionable insights.

Smarter Homes

Home automation elements are being manufactured in plenty, linking of distributed and disparate devices are being smoothened out, conceiving newer and nimbler services are in full swing, enabling frameworks, infrastructures are virtualized and pooled, and proven processes are in place, clustered, brokered, federated, and cloud architectures are being worked out, and so on.

Smarter Grids

Due to vociferous demands, the modern power grid is transforming into a cyber-physical systems (CPS), where the physical infrastructure and cyber infrastructure must coordinate to ensure an efficient and reliable power energy grid. However, there are some practical challenges. Existing grid operations require human decisions. Also, renewables such as wind and solar energy are inherently unreliable and cause the electricity supply to be susceptible to the vagaries of nature.

IoT and Cloud for Smarter Cities

Technological innovations such as extremely and deeply connected devices, distributed, disposable, and diminutive sensors and actuators, and the emergence of the Internet as the open, public and affordable WAN communication infrastructure, clouds as the core, centralized and cognitive one-stop platform for hosting software applications and services, and so on, are enabling cities to capture valuable data, develop and deploy new services, and enhance existing services substantially, ushering in the era of smarter cities. These services can bring definite improvement in the effectiveness of city management, generate new growth opportunities for businesses, empower innovations in all aspects, and raise the quality of citizens’ lives.

Sensor – Cloud Integration for Smarter Cities

Sensors and actuators are the eyes and ears of future IT in order to effortlessly capture disparate data from an increasing array of distinct sources. Correspondingly there are cloud-based analytical solutions to incredibly capitalize the accumulated data to extract actionable knowledge toward achieving and sustaining the smarter world vision. With clouds emerging as the web scale, enterprise-class, highly organized, and optimized IT infrastructures, there is a persistent and pressing need for sensors at the ground to be seamlessly and spontaneously integrated with online, on-demand, and on- or off-premise cloud environments

Social and Sensor Data Fusion in Cloud

Mobile phones are not only enabled with communication capabilities but also blessed with more computing power these days. Further on, a variety of minuscule sensors are smartly embedded inside mobile phones to make them multifaceted in their offerings and outputs.

Smarter Health Care

The need is to empower bed-ridden, debilitated, and diseased people to lead an independent and digitally assisted living. For example, if a person living alone wants to have a cup of filtered coffee, he or she can instruct the coffee maker in the kitchen through a smartphone command. Once the coffee is ready, a robot can be ordered to fetch the prepared coffee from the kitchen to his or her bedside. Similarly, medicine cabinets can have an alert or reminder facility to take tablets on time, the movement of the person can be monitored remotely, household items can be bought over through mobile commerce sites, food stuffs can be purchased online, all kinds of consumer electronics, energy managers, kitchen vessels and wares, instruments, micro ovens, infotainment and edutainment systems, dishwashers, toasters, refrigerators, and health care monitors, and so on can be instrumented and interconnected to have intelligent behaviours, thereby the everyday activities and requirements can be met with all choice and convenience

Cloud Inspired Smarter Health Care Services

The unprecedented adoption of the cloud concepts by various business verticals has impacted the health care industry too. The health care industry's technology infrastructure has been highly fragmented, inflexible, closed, and expensive. Due to the widely expressed concerns of data security, IT platforms, and health care applications and data are overwhelmingly maintained in-house IT infrastructures. Now having understood the strategic benefits, the health care sector across the world is keen to systematically leverage the cost and agility benefits of the cloud paradigm without compromising on the aspect of data security.

15. a. Explain the steps involved in creation of the following
1) AWS IoT – Policy **(8 marks)**

Create an AWS IoT policy that allows your Raspberry Pi to connect and send messages to AWS IoT.

1) In the AWS IoT console, if a Get started button appears, choose it. Otherwise, in the navigation pane, expand Security, and then choose Policies.

2) If a You don't have any policies yet dialog box appears, choose Create a policy. Otherwise, choose Create.

3) Enter a name for the AWS IoT policy (for example, MoistureSensorPolicy).

4) In the Add statements section, replace the existing policy with the following JSON. Replace region and account with your AWS Region and AWS account number.

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Effect": "Allow",
      "Action": "iot:Connect",
      "Resource": "arn:aws:iot:region:account:client/RaspberryPi"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Publish",
      "Resource": [

"arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/update
",

"arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/delete"
,

"arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/get"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": [

"arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/update
/accepted",

"arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/delete/
accepted",

"arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/get/acc
epted",

"arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/update
/rejected",

"arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/delete/
rejected"
      ]
    },
```

```
    {
      "Effect": "Allow",
      "Action": "iot:Subscribe",
      "Resource": [

"arn:aws:iot:region:account:topicfilter/$aws/things/RaspberryPi/shadow/update/accepted",

"arn:aws:iot:region:account:topicfilter/$aws/things/RaspberryPi/shadow/delete/accepted",

"arn:aws:iot:region:account:topicfilter/$aws/things/RaspberryPi/shadow/get/accepted",

"arn:aws:iot:region:account:topicfilter/$aws/things/RaspberryPi/shadow/update/rejected",

"arn:aws:iot:region:account:topicfilter/$aws/things/RaspberryPi/shadow/delete/rejected"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:GetThingShadow",
        "iot:UpdateThingShadow",
        "iot:DeleteThingShadow"
      ],
      "Resource": "arn:aws:iot:region:account:thing/RaspberryPi"

    }
  ]
}
```

5. Choose **Create**.

2)AWS IoT – Thing Object **(8 marks)**

Create a thing in the AWS IoT registry to represent your Raspberry Pi.

1. In the AWS IoT console, in the navigation pane, choose Manage, and then choose Things.

2. If a You don't have any things yet dialog box is displayed, choose Register a thing. Otherwise, choose Create.

3. On the Creating AWS IoT things page, choose Create a single thing.

4. On the Add your device to the device registry page, enter a name for your IoT thing (for example, RaspberryPi), and then choose Next. You can't change the name of a thing after you create it. To change a thing's name, you must create a new thing, give it the new name, and then delete the old thing.

5. On the Add a certificate for your thing page, choose Create certificate.

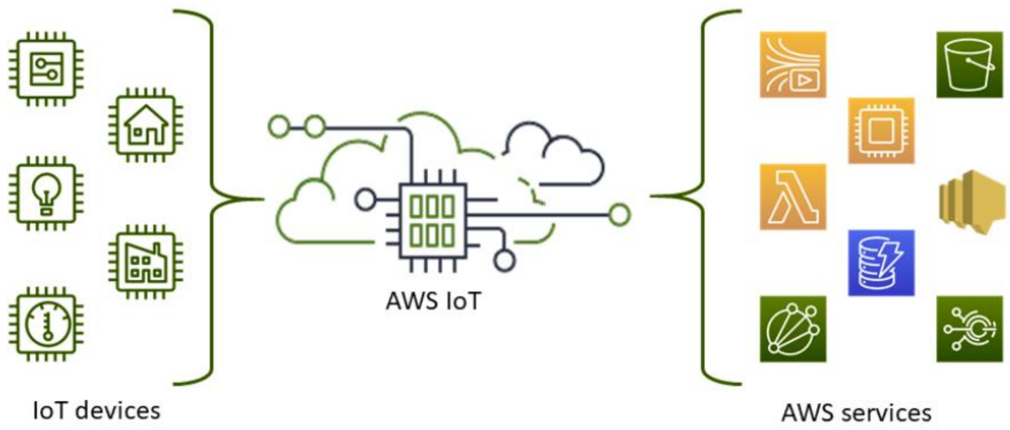
6. Choose the Download links to download the certificate, private key, and root CA certificate.

7. To activate the certificate, choose **Activate**. The certificate must be active for a device to connect to AWS IoT.
8. Choose **Attach a policy**.
9. For **Add a policy for your thing**, choose **MoistureSensorPolicy**, and then choose **Register Thing**.

Or

15.b. Briefly explain the interfaces available for devices and apps to access AWS IoT.(16)

Diagram 4 marks



4 marks

AWS IoT lets you select the most appropriate and up-to-date technologies for your solution. To help you manage and support your IoT devices in the field, AWS IoT Core supports these protocols:

- MQTT (Message Queuing and Telemetry Transport)
- MQTT over WSS (Websockets Secure)
- HTTPS (Hypertext Transfer Protocol - Secure)
- LoRaWAN (Long Range Wide Area Network)

(8 marks)

AWS IoT Device SDKs—Build applications on your devices that send messages to and receive messages from AWS IoT. For more information, see AWS IoT Device SDKs, Mobile SDKs, and AWS IoT Device Client.

AWS IoT Core for LoRaWAN—Connect and manage your long range WAN (LoRaWAN) devices and gateways by using AWS IoT Core for LoRaWAN.

AWS Command Line Interface (AWS CLI)—Run commands for AWS IoT on Windows, macOS, and Linux. These commands allow you to create and manage thing objects, certificates, rules, jobs, and policies. To get started, see the AWS Command Line Interface User Guide. For more information about the commands for AWS IoT, see iot in the AWS CLI Command Reference.

AWS IoT API—Build your IoT applications using HTTP or HTTPS requests. These API actions allow you to programmatically create and manage thing objects, certificates, rules, and policies. For more information about the API actions for AWS IoT, see Actions in the AWS IoT API Reference.

AWS SDKs—Build your IoT applications using language-specific APIs. These SDKs wrap the HTTP/HTTPS API and allow you to program in any of the supported languages.