Janvi Mineshbhai Patel

SUBJECT CODE : 3171108

As per New Syllabus of

GUJARAT TECHNOLOGICAL UNIVERSITY

Semester - VII(Electrical / IT) Open Elective - III

Semester - VII(ECE) Professional Elective - V

INTERNET OF THINGS

Iresh A. Dhotre

M.E. (Information Technology)

Ex-Faculty, Sinhgad College of Engineering,

Pune.

TECHNICAL PUBLICATIONS

SINCE 1993 An Up-Thrust for Knowledge

Scanned by TapScanner

TABLE OF CONTENTS

Scanned by TapScanner

Scanned by TapScanner

Scanned by TapScanner

Scanned by TapScanner

Scanned by TapScanner

1 Introduction to Internet of Things

Syllabus

IoT Definition, IoT characteristics, M2M and IoT, End to End IoT Architecture, Physical design of IoT, Logical Design of IoT, Overview of IoT protocols, IoT levels and deployment templates, Challenges for IoT, Interdependencies of IoT and cloud computing, Web of things

Contents

Scanned by TapScanner

1.1 Evolution of IoT

- The Internet of Things (IoT) refers to the capability of everyday devices to connect to other devices and people through the existing Internet infrastructure. Devices connect and communicate in many ways.
- Examples of this are smartphones that interact with other smartphones, vehicle-to-vehicle communication, connected video cameras, and connected medical devices.
- They are able to communicate with consumers, collect and transmit data to companies, and compile large amounts of data for third parties.
- Things are objects of the physical world (physical things) or of the information world (virtual world) which are capable of being identified and integrated into communication networks. Things have associated information, which can be static and dynamic.
- Physical things exist in the physical world and are capable of being sensed, actuated and connected. Examples of physical things include the surrounding environment, industrial robots, goods and electrical equipment.
- Virtual things exist in the information world and are capable of being stored, processed and accessed. Examples of virtual things include multimedia content and application software.
- A device is a piece of equipment with the mandatory capabilities of communication and optional capabilities of sensing, actuation, data capture, data storage and data processing.
- The devices collect various kinds of information and provide it to the information and communication networks for further processing. Some devices also execute operations based on information received from the information and communication networks.
- Fig. 1.1.1 shows evolutionary phase of internet.

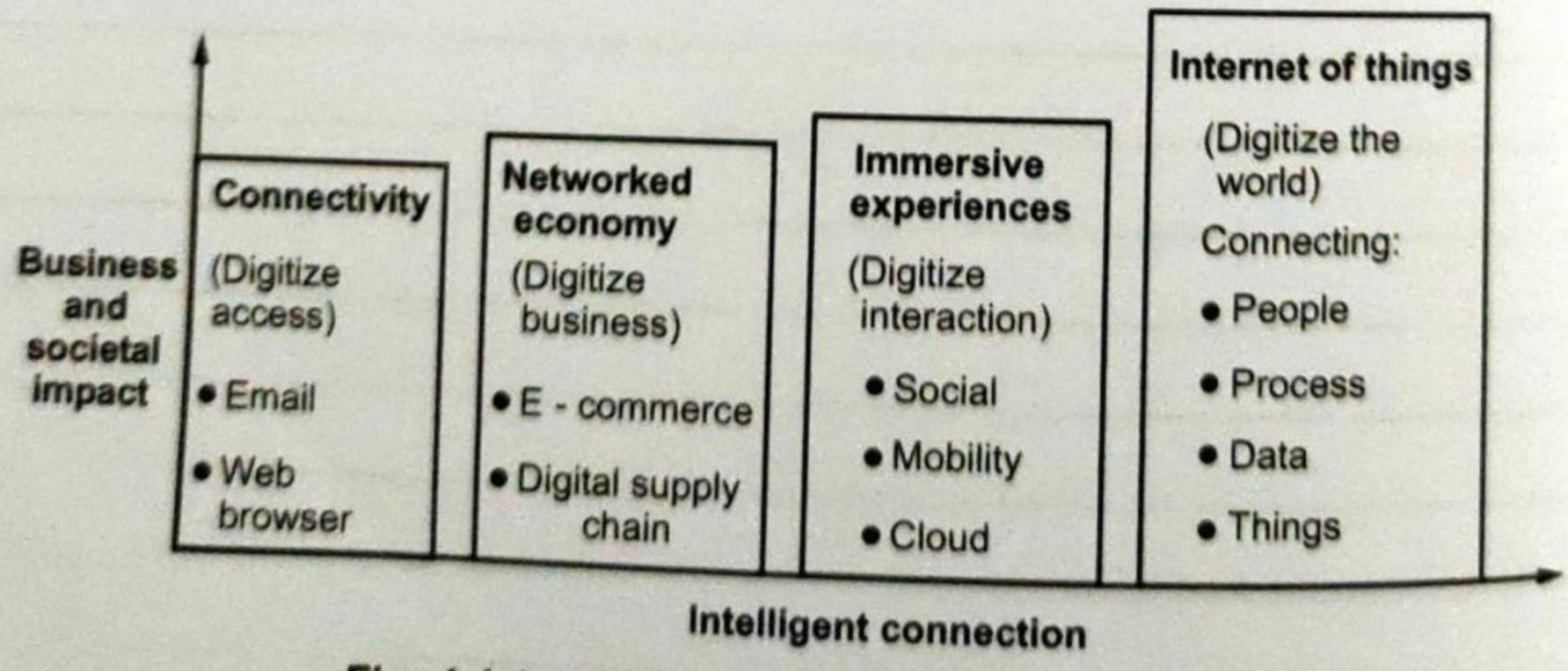


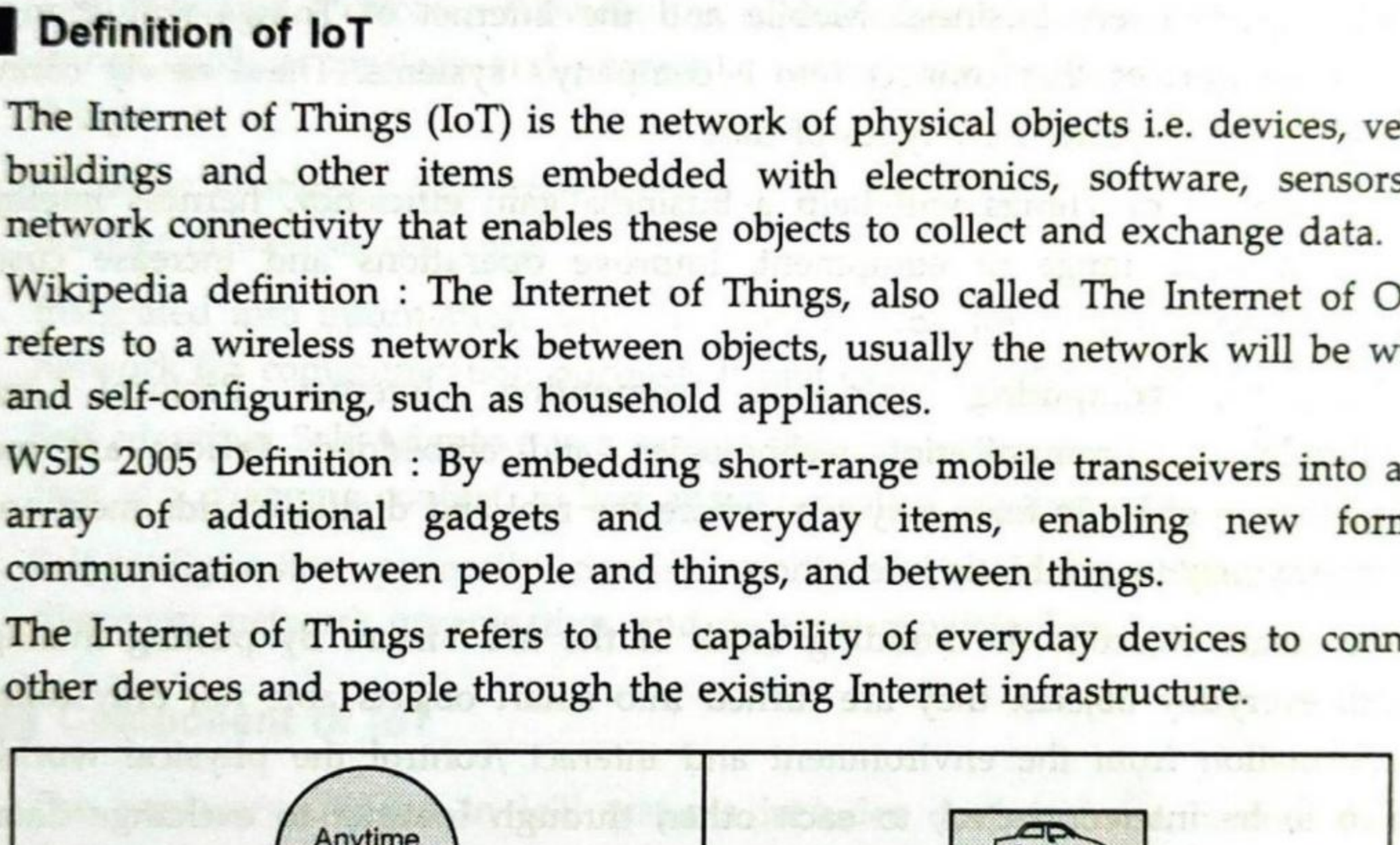
Fig. 1.1.1 : Evolutionary phase of internet

Scanned by TapScanner

- Evolutionary phase of internet is Connectivity, Networked Economy, Immersive Experiences and IoT.
  1. Connectivity : in the phase, peoples are connected to email, web services and searches the information.
  2. Networked Economy : this phase support e-commerce and supply chain enhancements along with collaborative engagement to drive increased efficiency in business processes.
  3. Immersive Experiences: this phase extended the Internet experience to encompass widespread video and social media while always being connected through mobility.
  4. Internet of Things: it adds connectivity to objects and machines in the world around us to enable new services and experiences.

1.1.1 Definition of IoT

- The Internet of Things (IoT) is the network of physical objects i.e. devices, vehicles, buildings and other items embedded with electronics, software, sensors, and network connectivity that enables these objects to collect and exchange data.
- Wikipedia definition : The Internet of Things, also called The Internet of Objects, refers to a wireless network between objects, usually the network will be wireless and self-configuring, such as household appliances.
- WSIS 2005 Definition : By embedding short-range mobile transceivers into a wide array of additional gadgets and everyday items, enabling new forms of communication between people and things, and between things.
- The Internet of Things refers to the capability of everyday devices to connect to other devices and people through the existing Internet infrastructure.



Scanned by TapScanner

- Devices connect and communicate in many ways. Examples of this are smart phones that interact with other smart phones, vehicle-to-vehicle communication, connected video cameras, and connected medical devices. They are able to communicate with consumers, collect and transmit data to companies, and compile large amounts of data for third parties.
- IoT data differs from traditional computing. The data can be small in size and frequent in transmission. The number of devices, or nodes, that are connecting to the network are also greater in IoT than in traditional PC computing.
- Machine-to-Machine communications and intelligence drawn from the devices and the network will allow businesses to automate certain basic tasks without depending on central or cloud based applications and services.
- IoT impacts every business. Mobile and the Internet of Things will change the types of devices that connect into a company's systems. These newly connected devices will produce new types of data.
- The Internet of Things will help a business gain efficiency, harness intelligence from a wide range of equipment, improve operations and increase customer satisfaction.
- Ubiquitous computing, pervasive computing, Internet Protocol, sensing technologies, communication technologies, and embedded devices are merged together in order to form a system where the real and digital worlds meet and are continuously in symbiotic interaction.
- The smart object is the building block of the IoT vision. By putting intelligence into everyday objects, they are turned into smart objects able not only to collect information from the environment and interact /control the physical world, but also to be interconnected, to each other, through Internet to exchange data and information.
- The expected huge number of interconnected devices and the significant amount of available data open new opportunities to create services that will bring tangible benefits to the society, environment, economy and individual citizens.
- However, the IoT is still maturing, in particular due to a number of factors, which limit the full exploitation of the IoT. Some of the factors are listed below :
  1. There is no unique identification number system for object in the world.
  2. IoT uses Architecture Reference Model (ARM) but there is no further development in ARM.

Scanned by TapScanner

3. Missing large-scale testing and learning environments
4. Difficulties in exchanging of sensor information in heterogeneous environments.
5. Difficulties in developing business which embraces the full support of the Internet of Things.

1.1.2 IoT Characteristics

1. Interconnectivity : Everything can be connected to the global information and communication infrastructure.
2. Heterogeneity : Devices within IoT have different hardware and use different networks but they can still interact with other devices through different networks.
3. Things-related services : Provides things-related services within the constraints of things, such as privacy and semantic consistency between physical and virtual thing.
4. Dynamic changes : The state of a device can change dynamically, thus the number of devices can vary.
5. Integrated into information network : IoT devices are integrated with information network for communication purpose. It will exchange data with other devices.
6. Self-adapting: Self-Adaptive is a system that can automatically modify itself in the face of a changing context, to best answer a set of requirements.
7. Self-configuration primarily consists of the actions of neighbour and service discovery, network organization, and resource provisioning.

1.1.3 Component of IoT

- The hardware utilized in IoT systems includes devices for a remote dashboard, devices for control, servers, a routing or bridge device, and sensors. These devices manage key tasks and functions such as system activation, action specifications, security, communication, and detection to support-specific goals and actions.
- Major components of IoT devices are as follows :

1. **Control units** : A small computer on a single integrated circuit containing processor core, memory and a programmable I/O peripheral. It is responsible for the main operation.
2. **Sensor :** Devices that can measure a physical quantity and convert it into a signal, which can be read and interpreted by the microcontroller unit. These devices consist of energy modules, power management modules, RF modules, and sensing modules. Most sensors fall into 2 categories : Digital or analog. An analog data is converted to digital value that can be transmitted to the Internet.

Scanned by TapScanner

a. Temperature sensors : accelerometers
b. Image sensors : gyroscopes
c. Light sensors : acoustic sensors
d. Micro flow sensors : humidity sensors
e. Gas RFID sensors : pressure sensors

3. **Communication modules :** These are the part of devices and responsible for communication with rest of IoT platform. They provide connectivity according to wireless or wired communication protocol they are designed. The communication between IoT devices and the Internet is performed in two ways :

a) There is an Internet-enable intermediate node acting as a gateway;

b) The IoT Device has direct communication with the Internet.

- The communication between the main control unit and the communication module uses serial protocol in most cases.

4. **Power Sources :** In small devices the current is usually produced by sources like batteries, thermocouples and solar cells. Mobile devices are mostly powered by lightweight batteries that can be recharged for longer life duration.

- **Communication Technology and Protocol :** IoT primarily exploits standard protocols and networking technologies. However, the major enabling technologies and protocols of IoT are RFID, NFC, low-energy Bluetooth, low-energy wireless, low-energy radio protocols, LTE-A, and WiFi-Direct. These technologies support the specific networking functionality needed in an IoT system in contrast to a standard uniform network of common systems.

1.1.4 Working of IoT

- Fig 1.1.2 shows working of IoT.

1. **Collect and transmit data :** The device can sense the environment and collect information related to it and transmit it to a different device or to the Internet.

2. **Actuate device based on triggers :** It can be programmed to actuate other devices based on conditions set by user.

3. **Receive information :** Device can also receive information from the network.

4. **Communication assistance :** It provides communication between two devices of same network or different network.

Scanned by TapScanner

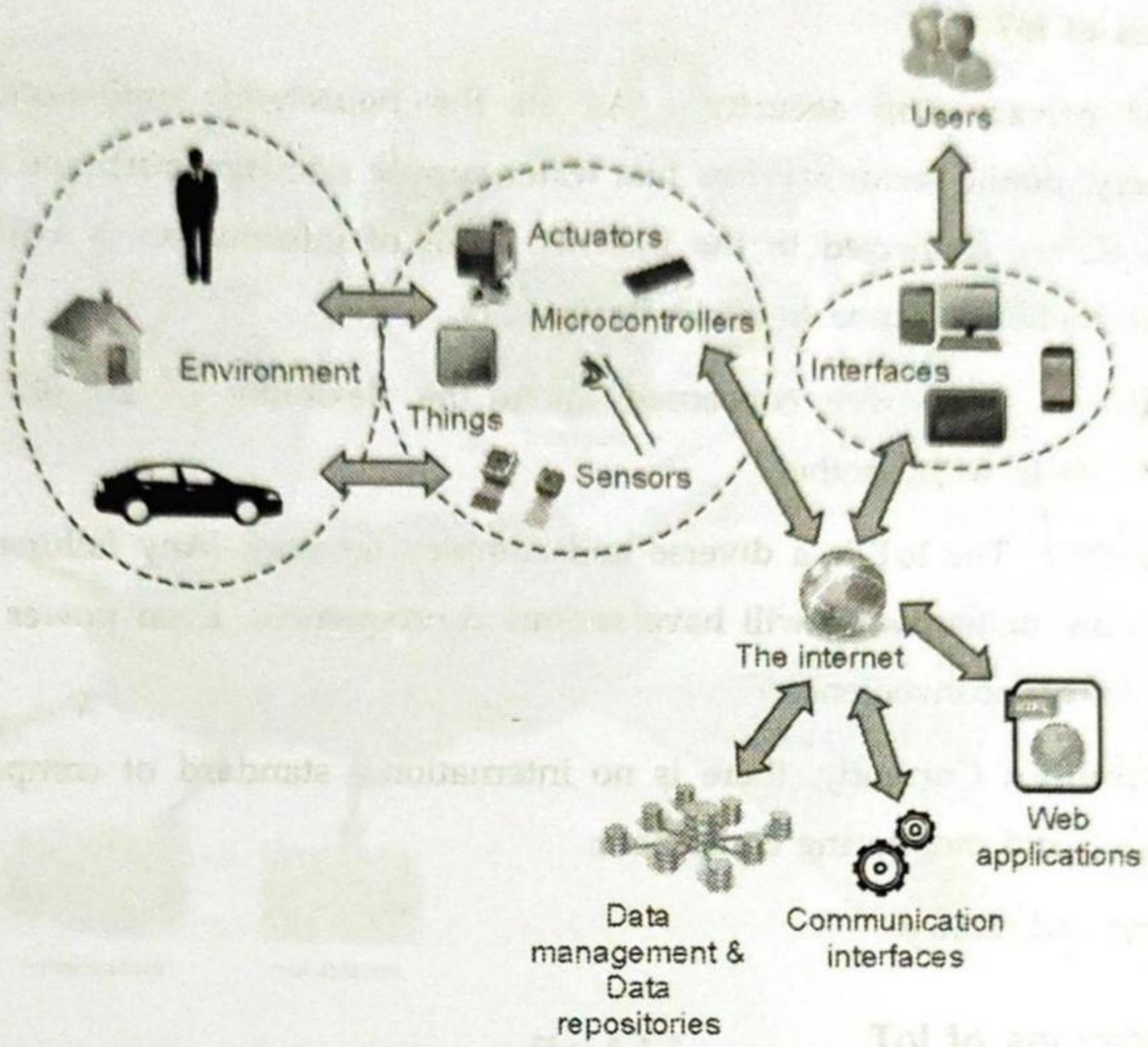


Fig. 1.1.2 : Working of IoT

- Sensors for various applications are used in different IoT devices as per different applications such as temperature, power, humidity, proximity, force etc.
- Gateway takes care of various wireless standard interfaces and hence one gateway can handle multiple technologies and multiple sensors.
- The typical wireless technologies used widely are 6LoWPAN, Zigbee, Zwave, RFID, NFC etc. Gateway interfaces with cloud using backbone wireless or wired technologies such as WiFi, Mobile , DSL or Fibre.

1.1.5 Advantages and Disadvantages

Advantages of IoT

1. Improved customer engagement and communication.
2. Support for technology optimization
3. Support wide range of data collection
4. Reduced waste

Scanned by TapScanner

Disadvantages of IoT

1. **Loss of privacy and security :** As all the household appliances, industrial machinery, public sector services like water supply and transport, and many other devices all are connected to the Internet, a lot of information is available on it. This information is prone to attack by hackers.
2. **Flexibility :** Many are concerned about the flexibility of an IoT system to integrate easily with another.
3. **Complexity :** The IoT is a diverse and complex network. Any failure or bugs in the software or hardware will have serious consequences. Even power failure can cause a lot of inconvenience.
4. **Compatibility :** Currently, there is no international standard of compatibility for the tagging and monitoring equipment.
5. Save time and money.

1.1.6 Applications of IoT

1. **Home :** Buildings where people live. It controls home and security systems.
2. **Offices :** Energy management and security in office buildings; improved productivity, including for mobile employees.
3. **Factories :** Places with repetitive work routines, including hospitals and farms; operating efficiencies, optimizing equipment use and inventory.
4. **Vehicles :** Vehicles including cars, trucks, ships, aircraft, and trains; condition-based maintenance, usage-based design, pre-sales analytics
5. **Cities :** Public spaces and infrastructure in urban settings; adaptive traffic control, smart meters, environmental monitoring, resource management.
6. **Worksites :** It is custom production environments like mining, oil and gas, construction; operating efficiencies, predictive maintenance, health and safety.

Scanned by TapScanner

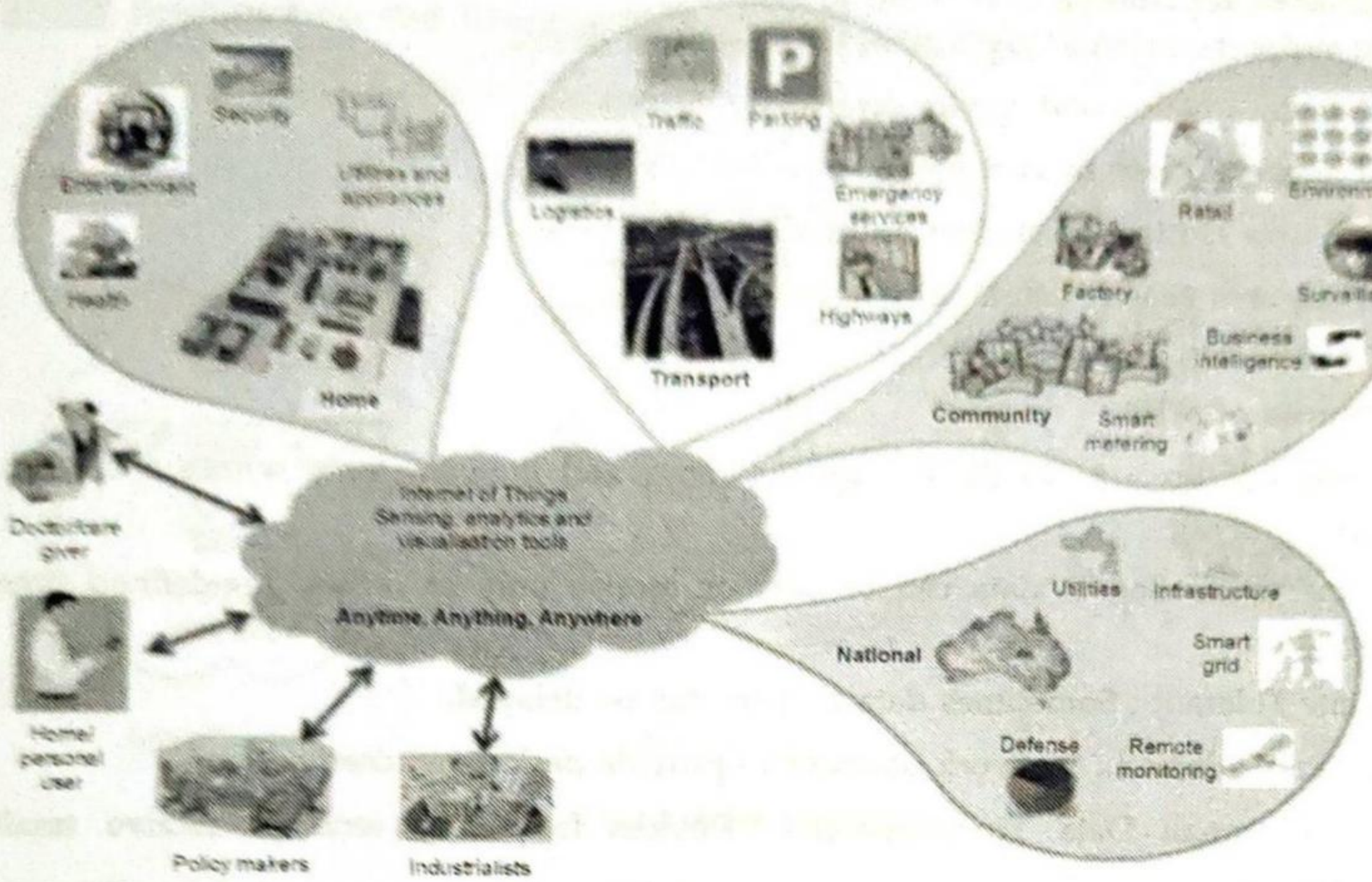


Fig. 1.1.3

1.2 M2M and IoT

- Machine to Machine (M2M) communication is the communication among the physical things which do not need human intervention.
- M2M communication is a form of data communication that involves one or more entities that do not necessarily require human interaction or intervention in the process of communication. M2M is also named as Machine Type Communication (MTC) in 3GPP.
- M2M communication could be carried over mobile networks (e.g. GSM-GPRS, CDMA EVDO networks). In the M2M communication, the role of mobile network is largely confined to serve as a transport network.
- M2M is only a subset of IoT. IoT is a more encompassing phenomenon because it also includes Human-to-Machine communication (H2M).
- Radio Frequency Identification (RFID), Location-Based Services (LBS), Lab-on-a-Chip (LOC), sensors, Augmented Reality (AR), robotics and vehicle telematics, which are some of the technology innovations that employ both M2M and H2M communications.

Scanned by TapScanner

- Reasons for shifting from M2M to IoT :

1. It supports multiple application with multiple device.
2. It is information and service centric.
3. It supports open market place.
4. IoT uses Horizontal enabler approach.
5. It requires generic commodity devices.
6. Used in B2B and B2C.

Key features of M2M :

1. Low Mobility : M2M Devices do not move and if moves only within a certain area.
2. Time Controlled : data can be sent or receive only at certain pre-defined time periods.
3. Time Tolerant : Sometimes data transfer can be delayed.
4. Packet Switched : Network operator to provide packet switched service
5. Online small Data Transmissions : Devices frequently send or receive small amounts of data.
6. Low Power Consumption : To improve the ability of the system to efficiently service M2M applications.
7. Location Specific Trigger : Intending to trigger M2M device in a particular area e.g. wake up the device.

Six Pillars of M2M :

- The six pillars of M2M are as follows :

1. Remote monitoring is a generic term most often representing supervisory control, data acquisition and automation of industrial assets.
2. RFID is a data-collection technology that uses electronic tags for storing data.
3. A sensor network monitors physical or environmental conditions, with sensor nodes acting cooperatively to form/maintain the network.
4. The term smart service refers to the process of networking equipment and monitoring it at a customer's site so that it can be maintained and serviced more effectively.
5. Telematics to the integration of telecommunications and infomatics, but most often it refers to tracking, navigation and entertainment applications in vehicles.
6. Telemetry is usually associated with industrial, medical and wildlife-tracking applications that transmit small amounts of vehicles data.

Scanned by TapScanner

1.2.1 Architecture and Components of M2M

- Fig. 1.2.1 shows M2M architecture.

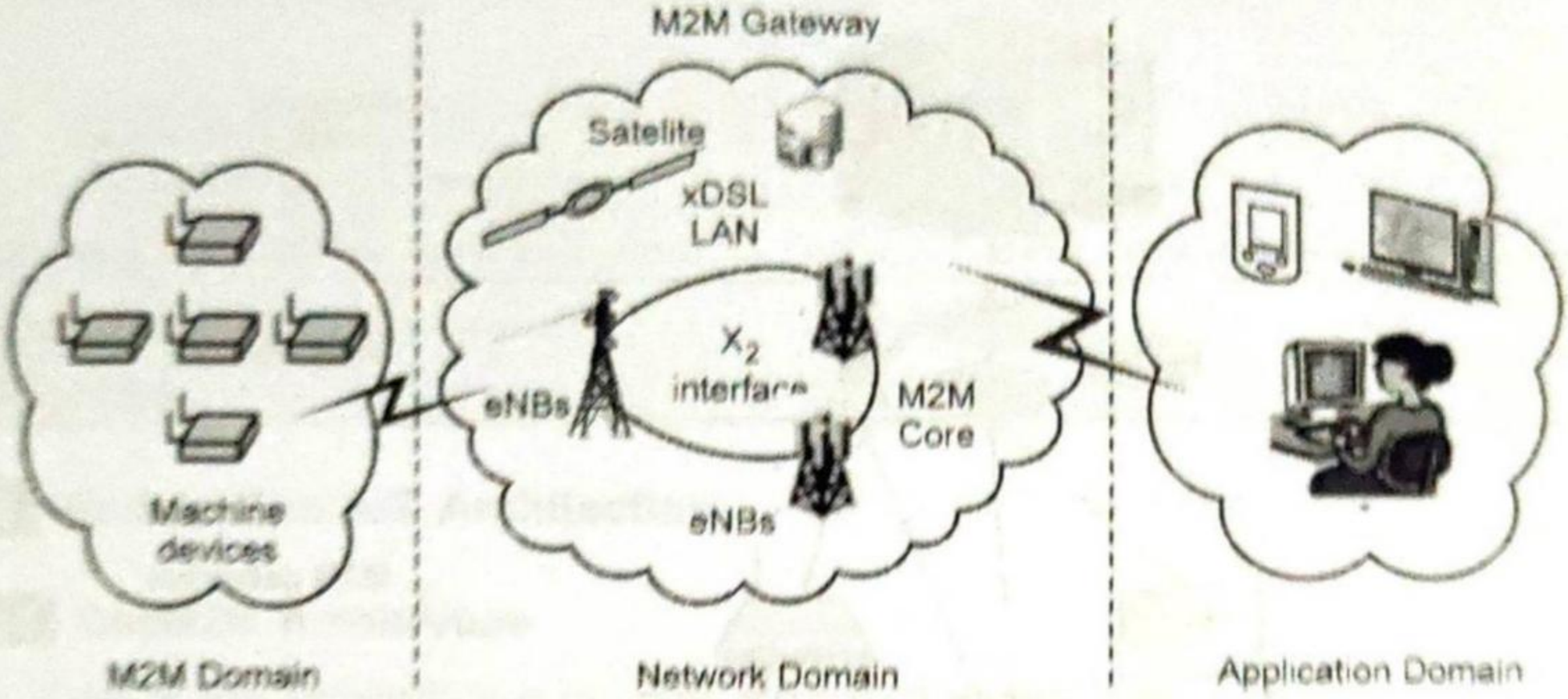


Fig. 1.2.1 : M2M architecture

- *The system components of an M2M solution are as follows :*

1. *M2M Device :* A device that runs application(s) using M2M capabilities and network domain functions. An M2M device is either connected straight to an access network or interfaced to M2M gateways via an M2M area network.

2. *M2M area network :* A M2M area network provides connectivity between M2M devices and M2M gateways. Examples of M2M area networks include : Personal area network technologies such as IEEE 802.15, 3RD, UWB, Zigbee, Bluetooth, etc or local networks such as PLC, M-BUS, Wireless M-BUS.

3. *M2M gateways :* Equipments using M2M capabilities to ensure M2M devices interworking and interconnection to the network and application domain. The M2M gateway may also run M2M applications.

4. *M2M applications server :* Applications that run the service logic and use service capabilities accessible via open interfaces.

5. *M2M application :* The application component of the solution is a realisation of the highly specific monitor and control process. The application is further integrated into the overall business process system of the enterprise.

Scanned by TapScanner

- Fig. 1.2.2 shows generic M2M solution.

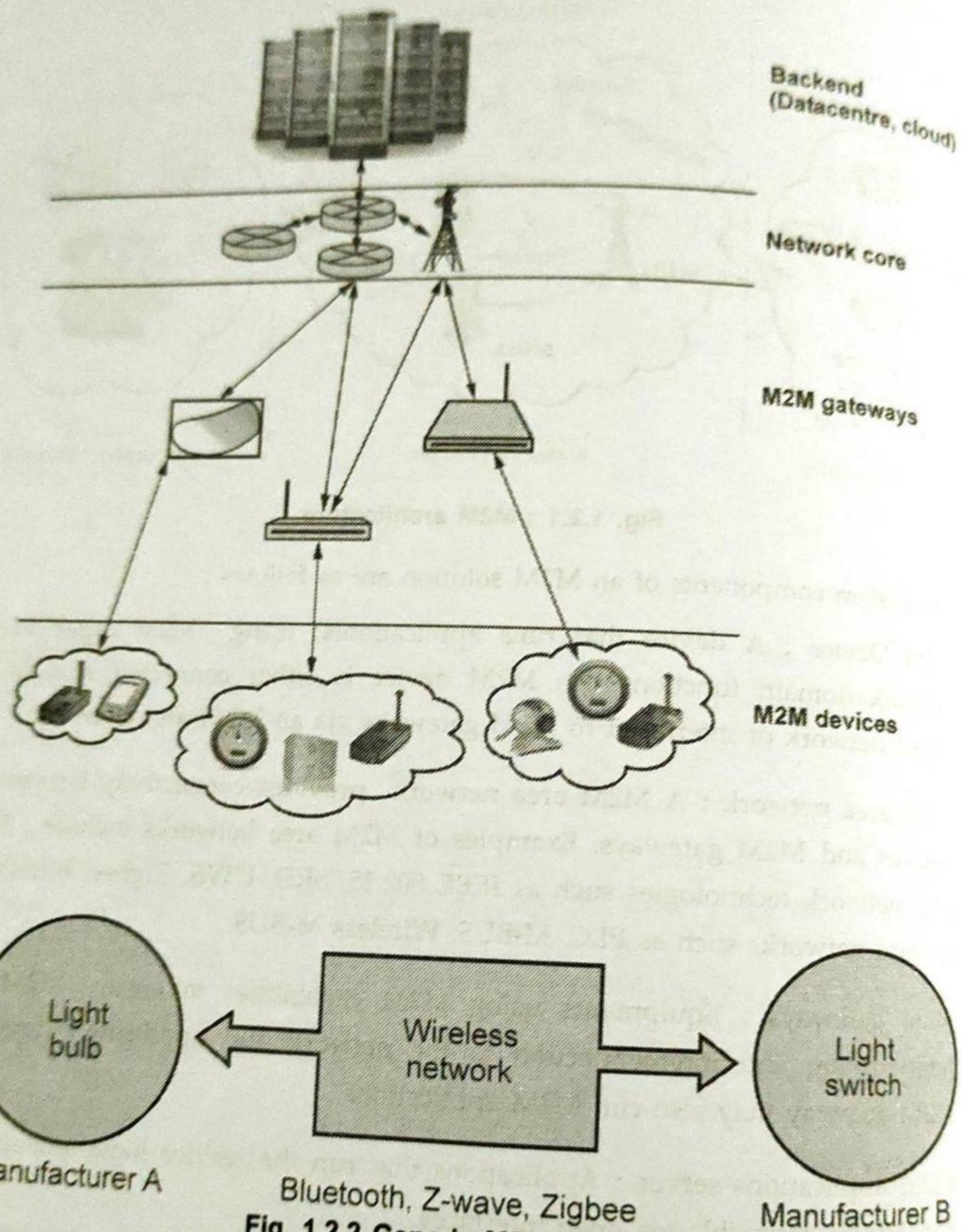


Fig. 1.2.2 Generic M2M solution

- A number of sub-sets of users of M2M services can be identified : Consumers in the home, business users and facility managers, city governments, logistics businesses, energy providers and more.

Scanned by TapScanner

1.2.2 Difference between M2M and IoT

Machine-to-Machine	Internet of Things
It support single application with single device.	It support multiple application with multiple device.
It is communication and device centric.	It is information and service centric.
It support closed business operations.	It support open market place.
M2M uses vertical system solution approach.	IoT uses horizontal enabler approach.
It requires specialized device solutions.	It requires generic commodity devices.
Used in B2B.	Used in B2B and B2C.

1.3 End to End IoT Architecture

1.3.1 OneM2M Architecture

- Goal of this architecture is to create a common service layer, which can be readily embedded I field devices to allow communication with application servers.
- Fig 1.3.1 shows OneM2M architecture. OneM2M architecture consists of application layer, services layer and network layer.

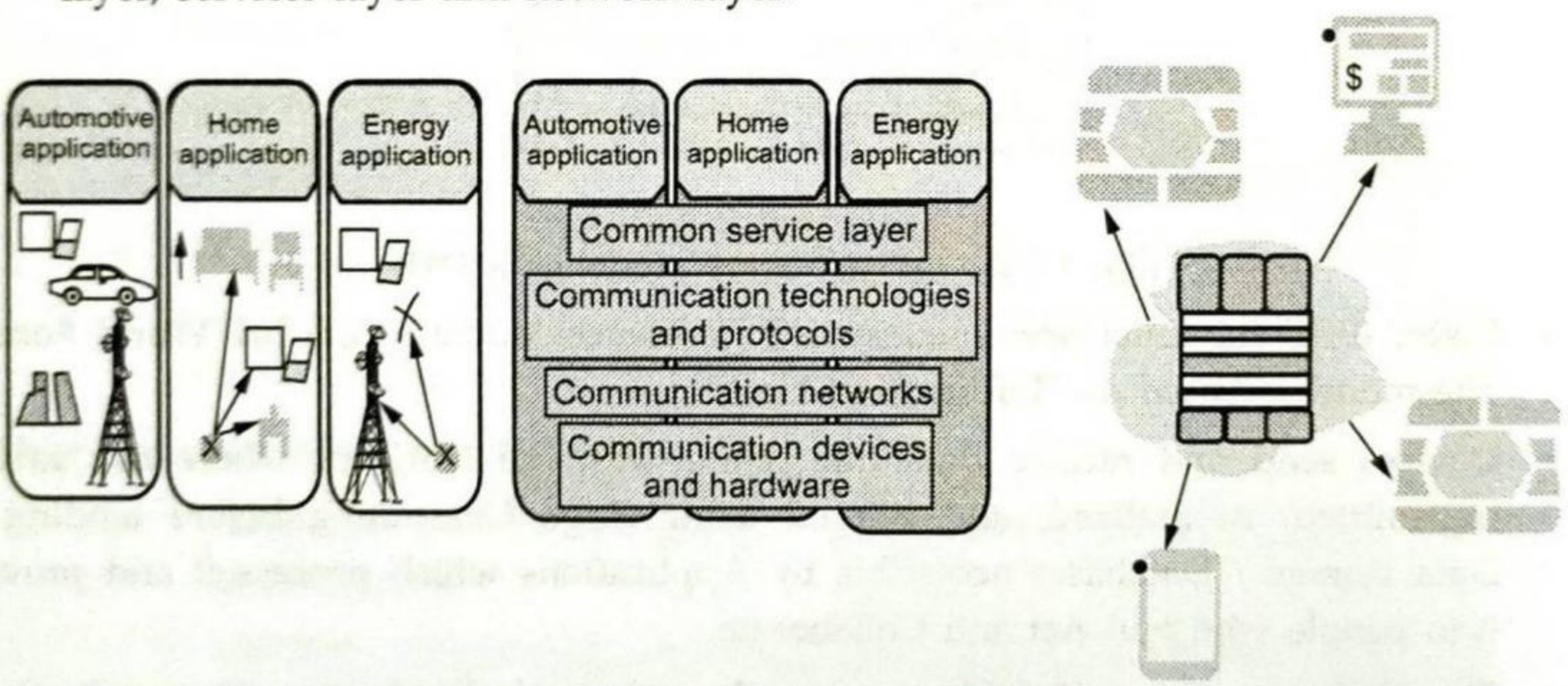


Fig. 1.3.1 : OneM2M architecture

1. Application layer : Comprises oneM2M application and related business and operational logic.
2. Services layer : Consists of OneM2M service function that enables oneM2M applications.
3. Network layer : It provides transport, connectivity and services functions.

Scanned by TapScanner

1.3.2 IoT World Forum Standardized Architecture

- Fig 1.3.2 shows IoT World Forum (IoTWF) Standardized Architecture.

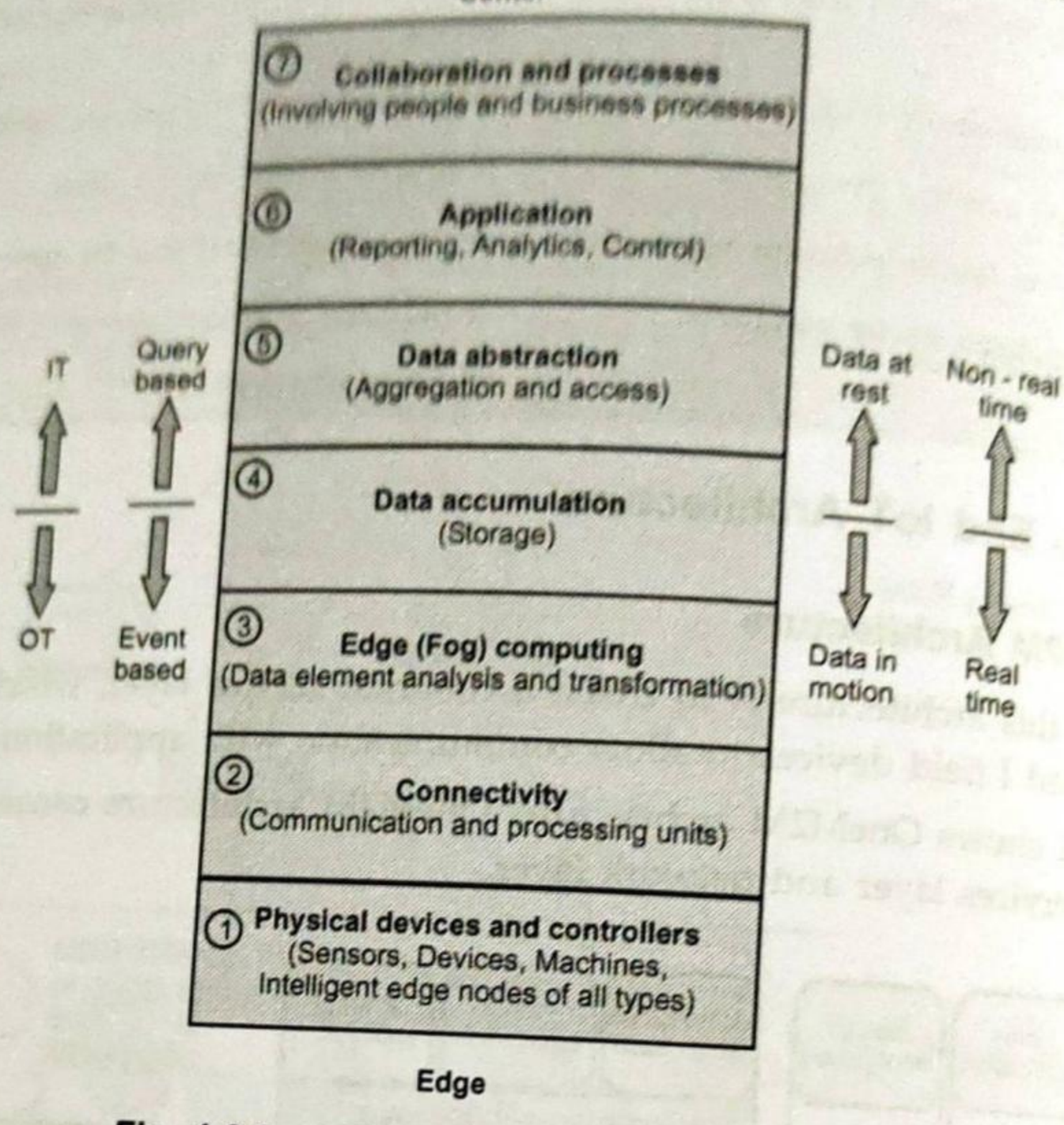


Fig. 1.3.2 : IoT reference model of IoTWF

- Cisco, IBM, and Intel presented an IoT Reference Model at the IoT World Forum. The model is based on "Information Flow".
- Devices send and receive data interacting with the Network where the data is transmitted, normalized, and filtered using Edge Computing before landing in Data storage / Databases accessible by Applications which process it and provide it to people who will Act and Collaborate.
- The IWF is concerned with the broader issue of developing the applications, middleware, and support functions for an enterprise based IoT.
- IoT reference model define a set of levels with control flowing from center to the edge which includes sensors, device, machines and other type of an intelligent nodes.
- Layer 1 : Comprises physical devices and controllers that might control multiple devices. This level enables devices to communicate with one another and to communicate, via the upper logical levels, with application platforms such as computers, remote-control devices, and smart-phones.

Scanned by TapScanner

- Layer 2 : The IWF model includes gateways in level 2. Because the gateway is a networking and connectivity device, its placement at level 2 seems to make more sense.
- Layer 3 : It performs data element analysis and transformation.
- Layer 4 : The data accumulation level, is where data coming from the numerous devices, and filtered and processed by the edge computing level, is placed in storage that will be accessible by higher levels. This level marks a clear distinction in the design issues, requirements, and method of processing between lower-level (fog) computing and upper-level (typically cloud) computing.
- Layer 5, Data abstraction layer : Reconciles multiple data formats and ensures consistent semantics from various sources. Confirms that data set is complete and consolidates data into one place or multiple data stores using virtualization.
- Layer 6, Application layer : Interprets data using software applications. Applications may monitor, control and provide reports based on the analysis of the data.
- Layer 7, Collaboration and processes layer : Consumes and shares the application information.
- Using this reference model, following things are achieved :
  1. Decompose the problem into smaller parts.
  2. Identify different technologies at each layer.
  3. Define a system in which different parts can be provided by different vendors.
  4. Have a process are defining interface that leads to interoperability.
  5. Define a tiered security model that is enforced at one transition points between levels.

Characteristics of IoTWF model :

- Simplifies : It helps break down complex systems so that each part is more understandable.
- Clarifies : It provides additional information to precisely identify levels of the IoT and to establish common terminology.
- Identifies : It identifies where specific types of processing are optimized across different parts of the system.
- Standardizes : It provides a first step in enabling vendors to create IoT products that work with each other.
- Organizes : It makes the IoT real and approachable, instead of simply conceptual.

Scanned by TapScanner

1.3.3 Simplified IoT Architecture

- Fig 1.3.3 shows Simplified IoT Architecture.

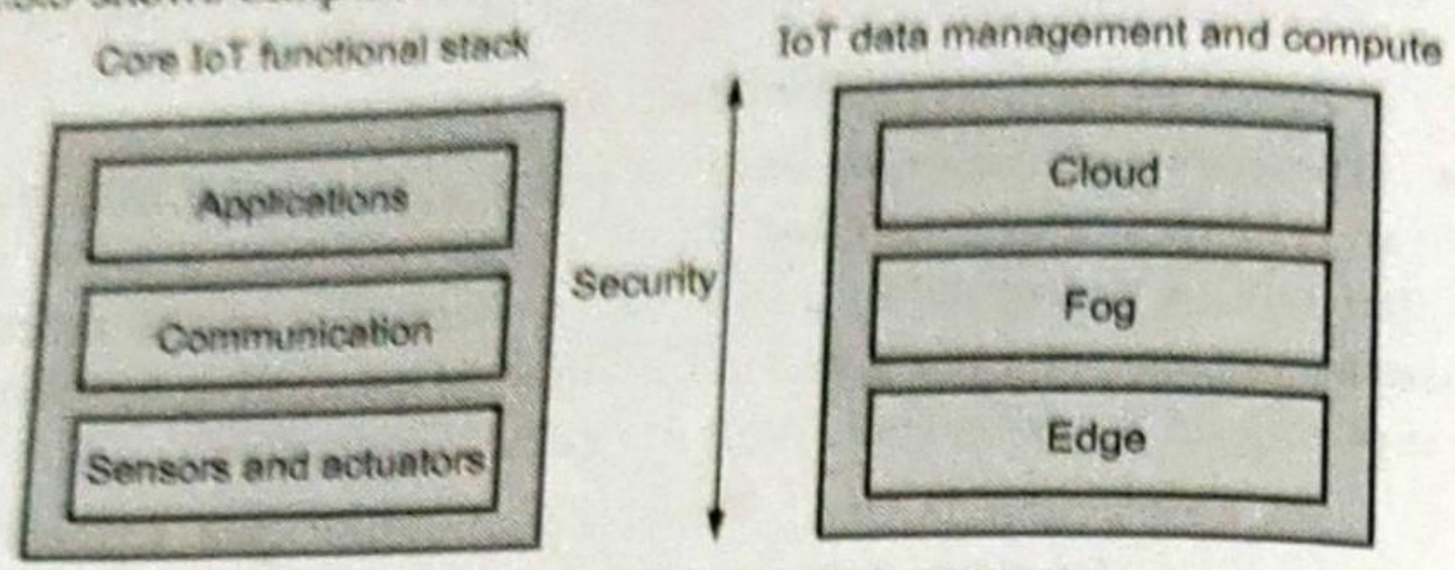


Fig. 1.3.3 : Simplified IoT Architecture

- It consists of Core IoT functional stack group and IoT data management and compute stack.

IoT data management and compute stack :

- Fog computing, or sometimes called edge computing, can be thought of as an extension of the cloud, with the infrastructure distributed at the edge of the network.
- Fog computing facilitates the operation of end devices, typically smart IoT devices, with cloud computing data centres.
- This helps in meeting the needs of high-speed mobile scenarios and geographical distribution scenarios and reduces the bandwidth load of the network core.
- The IoT Cloud Platform represents the software infrastructure and services required to enable an IoT solution.
- An IoT Cloud Platform typically operates on a cloud infrastructure or inside an enterprise data center and is expected to scale both horizontally, to support the large number of devices connected, as well as vertically to address the variety of IoT solutions.
- The IoT Cloud Platform will facilitate the interoperability of the IoT solution with existing enterprise applications and other IoT solutions.

Core IoT functional stack group :

- The IoT gateway acts as the aggregation point for a group of sensors and actuators to coordinate the connectivity of these devices to each other and to an external network.
- An IoT gateway can be a physical piece of hardware or functionality that is incorporated into a larger "Thing" that is connected to the network.

Scanned by TapScanner

- For example, an industrial machine might act like a gateway, and so might a connected automobile or a home automation appliance.
- An IoT gateway will often offer processing of the data "at the edge" and storage capabilities to deal with network latency and reliability.
- For device to device connectivity, an IoT gateway deals with the interoperability issues between incompatible devices.
- A typical IoT architecture would have many IoT gateways supporting masses of devices.

1.4 Physical Design of IoT

- Physical Design of IoT system refers to IoT Devices and IoT Protocols.

1.4.1 Things in IoT

- IoT devices have unique identity and they are refer as "things" in IoT. Device can perform remote sensing, actuating and monitoring. IoT devices can exchange data between them and process data or send to centralized location for processing and storage. Fig. 1.4.1 shows block diagram of IoT device.

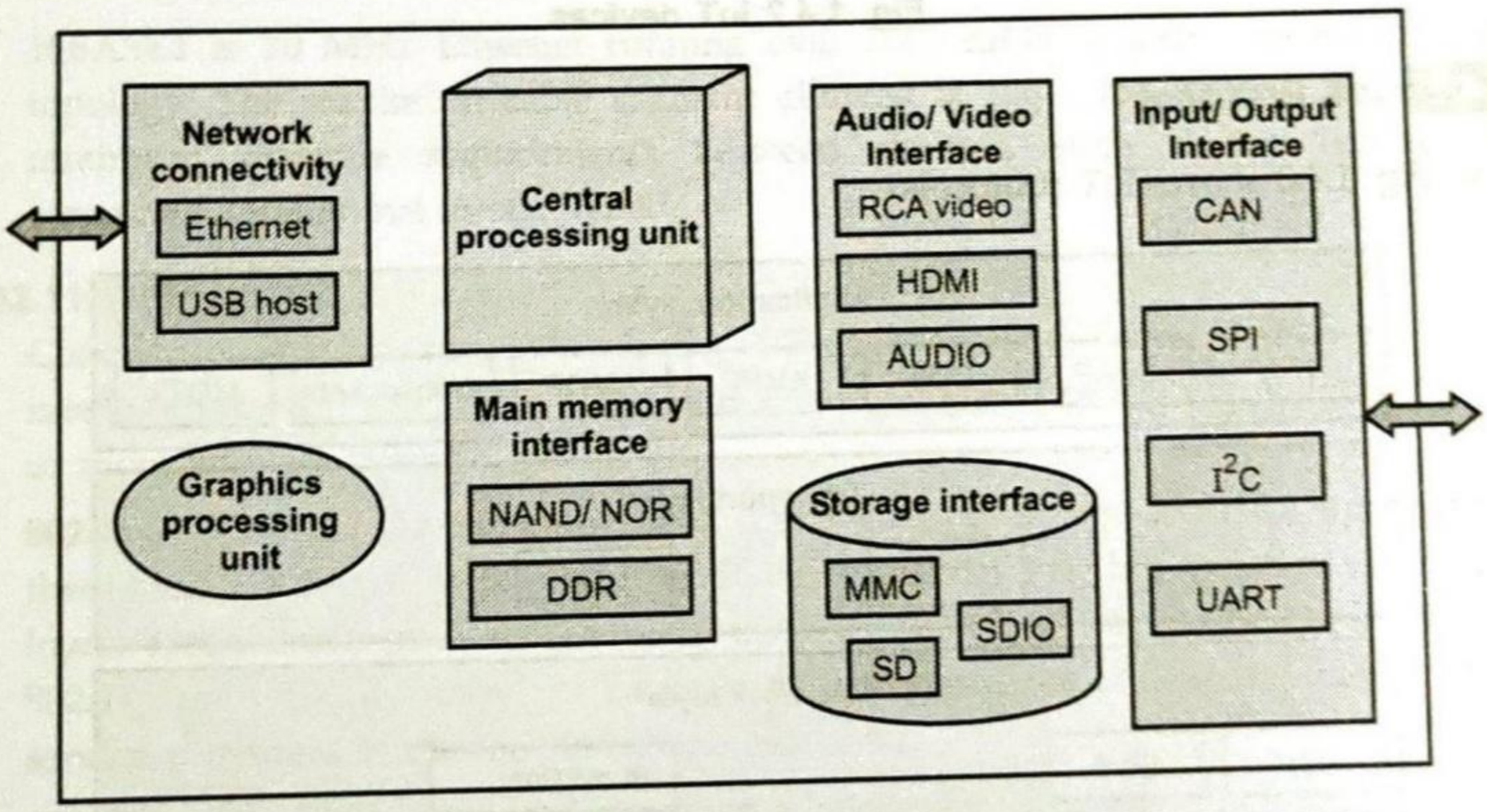


Fig. 1.4.1 Block diagram of IoT devices

- IoT devices provide interface to various wire and wireless devices. Interface includes memory interface, I/O interface for sensors, Internet connectivity interface, storage interface etc.

Scanned by TapScanner

- Using sensors, IoT collects various information like temperature, light intensity, humidity, air pressure. Some application used cloud based storage. Collected information is stored in cloud and transmitted to other devices.
- Various types of IoT devices are smart clothing, smart watch, wearable sensors, LED lights, automobile industry etc. Fig. 1.4.2 shows IoT devices.

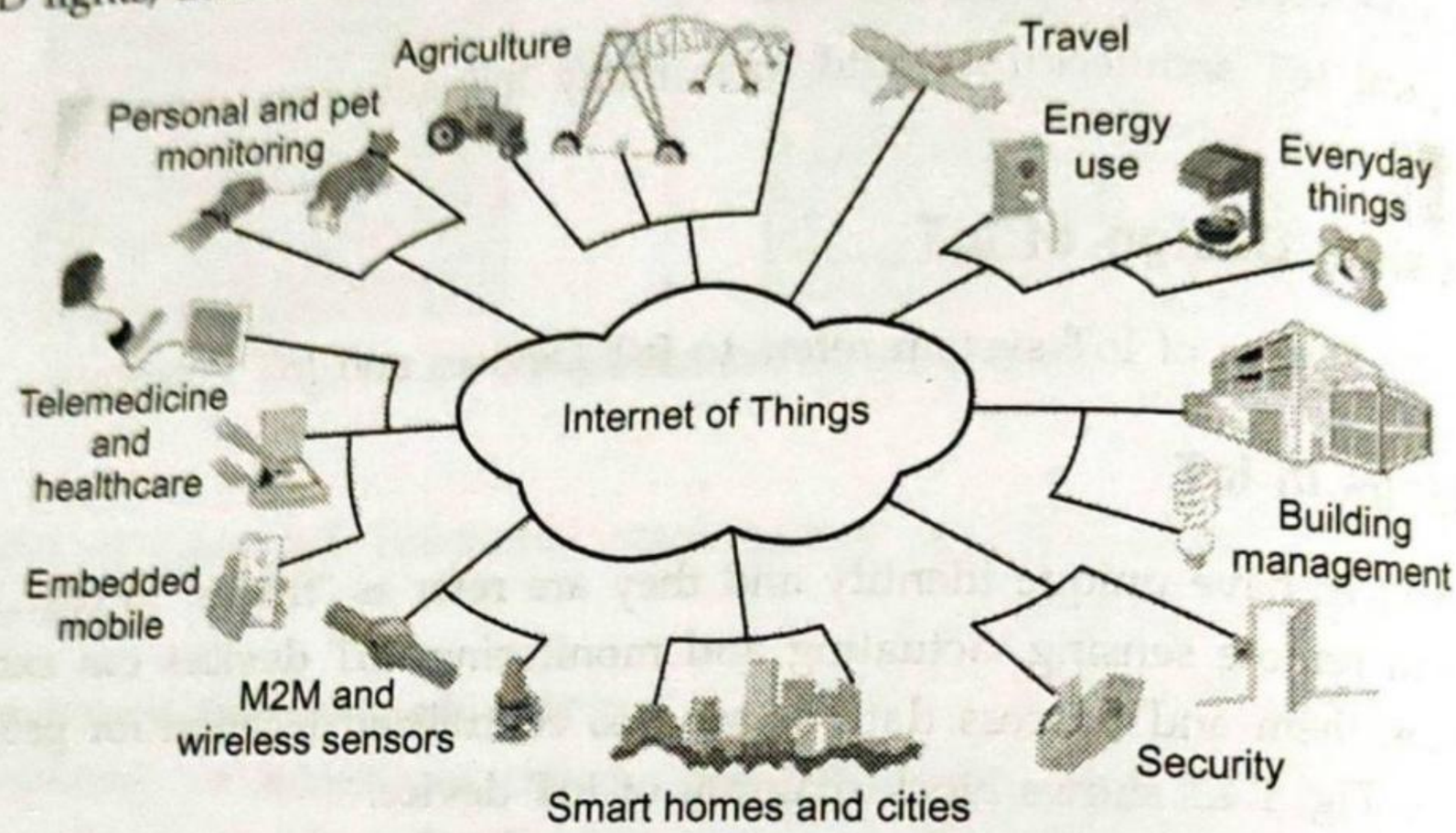


Fig. 1.4.2 IoT devices

1.4.2 IoT Protocol

- Fig. 1.4.3 shows IoT protocols.

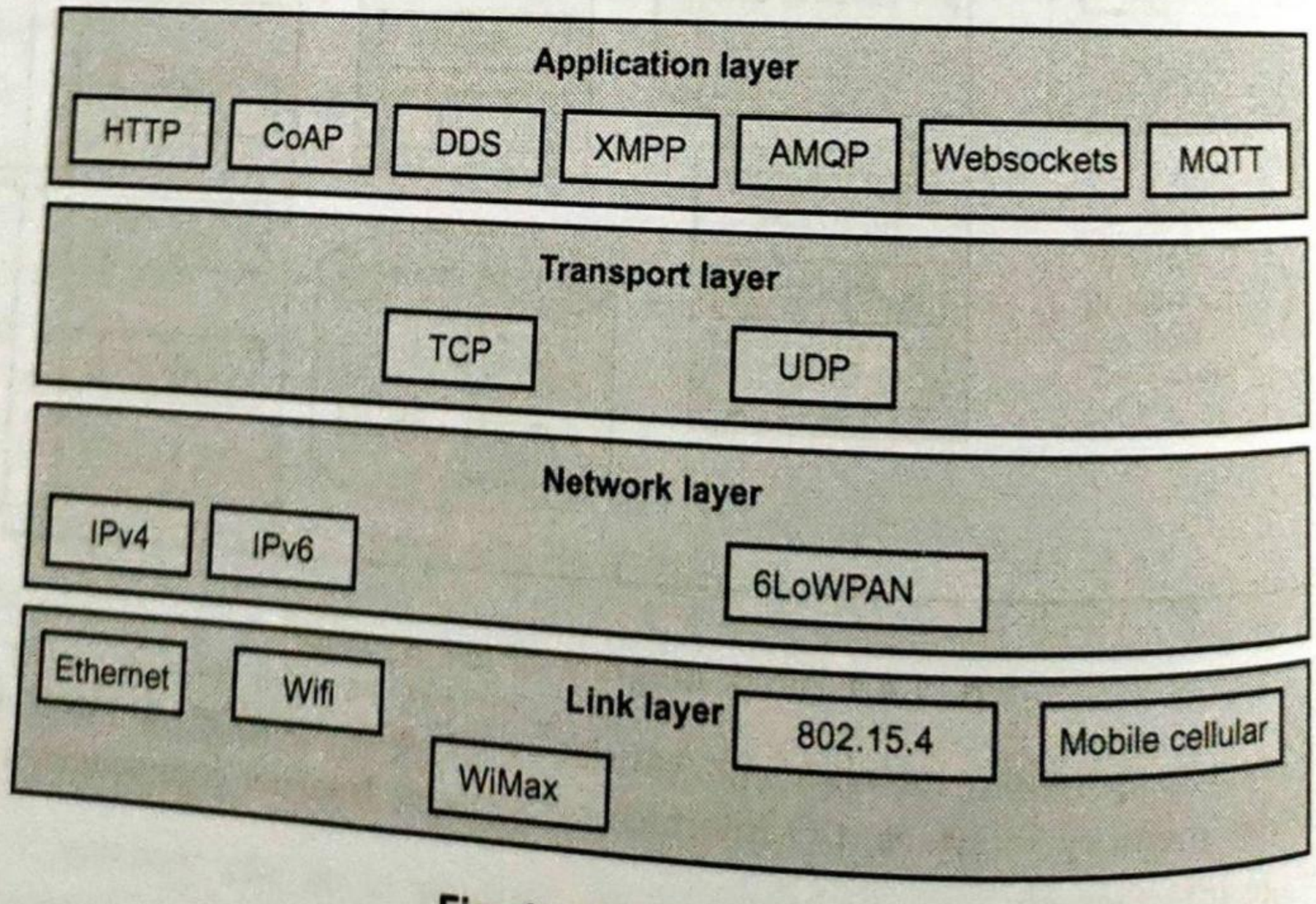


Fig. 1.4.3 IoT protocol

Scanned by TapScanner

1. Link Layer

- Link layer protocols decide how data is sent on physical medium. Link layer works within the local area network. Protocol of link layer is explained below :

a. 802.3 Ethernet

- This protocol is used for wired medium. Ethernet, in its most basic version runs at 10 Mbit/s. Ethernet has traditionally been used to network enterprise workstations and to transfer non-real-time data.
- The Ethernet standard allows for several different implementations such as twisted pair and coaxial cable. The maximum length of an Ethernet is determined by the nodes' ability to detect collisions.
- The worst case occurs when two nodes at opposite ends of the bus are transmitting simultaneously. Ethernet does not provide any mechanism for acknowledging received frames, making it what is known as an unreliable medium.
- Carrier sense multiple access with collision detection (CSMA/CD) is the most commonly used protocol for LANs. 10BASE5 is generally used as low cost alternative for fiber optic media for use as a backbone segment with in a single building.
- 10BASET is 10 MHz Ethernet running over UTP cable. It also uses passive star topology. The maximum cable segment allowed is 100 - 150 meters. There is no minimum distance requirements between devices, such devices cannot be connected serially but in star wired

b. 802.11 Wifi

- Commonly referred to as Wi-Fi the 802.11 standards define a through-the-air interface between a wireless client and a base station access point or between two or more wireless clients.
- **802.11a :** The 802.11a standard uses the 5 GHz spectrum and has a maximum theoretical 54 Mbps data rate. The 5GHz spectrum has higher attenuation than lower frequencies, such as 2.4 GHz used in 802.11b/g standards. Products with 802.11a are typically found in larger corporate networks or with wireless Internet service providers in outdoor backbone networks.
- **802.11b :** The 802.11 standard provides a maximum theoretical 11 Mbps data rate in the 2.4 GHz Industrial, Scientific and Medical (ISM) band.
- 802.11b uses Complementary Code Keying (CCK) instead of Differential Quadrature Phase Shift Keying (DQPSK) used at lower rates.
- **802.11g :** It provides 20 Mbps and more in the 2.4 GHz band.

Scanned by TapScanner

c. 802.16 WiMax

- WiMAX refers to broadband wireless networks that are based on the IEEE 802.16 standard, which ensures compatibility and interoperability between broadband wireless access equipment.
- The 802.16 astandard will support OFDM in the 2-to-11 GHz frequency range. The 802.16b standard will operate in the 5 GHz ISM band. A single WiMAX tower can provide coverage to a very large area as big as 3000 square miles
- WiMAX receiver : The receiver and antenna could be a small box or Personal Computer Memory card or they could be built into a laptop the way WiFi access is today.

d. 802.15.4 Zigbee

- In 2002, seeing that neither Wi-Fi nor Bluetooth could not fit some of their needs for embedded systems , a number of industrial companies formed the consortium called ZigBee Alliance, aimed at providing standards for low cost/low consumption wireless communications. Then, with the birth of IEEE 802.15.4 group.
- ZigBee communications can reach up to 500m, with a data rate of up to 250 kbs, for a typical power consumption of 125 to 400 μW.
- As ZigBee is based on IEEE 802.15.4, there is no wake-up signal, but slots for sleep or activity, or in asynchronous mode, devices sleeping anytime they have nothing to say, with an ever-vigilant coordinator.
- To use a ZigBee module with a microcontroller, you need to connect it to a UART. There are other, optional pins to use, including a number of analog inputs / digital IOs and a PWM output indicating the strength of the signal which you can directly connect to a LED pin for observation purposes.
- There are two modes of data transfer namely *Beacon mode* and *Non Beacon mode*.
- In Beacon mode, when the devices are not sending the data they may enter a low power state and reduces the power consumption.
- In Non-beacon mode, the end devices need to be wake up only while sending the data while the routers and coordinators need to be active most of the time.

e. Mobile Communication (2G/3G/4G)

- GSM frequencies originally designed on 900 MHz range, now also available on 800 MHz, 1800 MHz and 1900 MHz ranges. The backbone of a GSM network is a telephone network with additional cellular network capabilities
- 4G is also called as Long Term Evolution. It's promises data transfer rates of 100 Mbps.

Scanned by TapScanner

2. Network Layer

- The network layer is responsible for the delivery of packets from the source to destination.
- Network layer uses IP address to choose one host among millions of host. In network layer, datagram needs a destination IP address for delivery and a source IP address for a destination reply.

a. IPv4

- IP is used for communicating all Internet enabled devices. The transport layer is responsible for delivery of message from one process to another.
- The network does the host to destination delivery of individual packets considering it as independent packet. But transport layer ensures that the whole message arrives intact and in order with error control and process control.
- An IP address is a numeric identifier assigned to each machine on an IP network. IP address is a software address, not a hardware address, which is hard-coded in the machine or NIC.
- An IP address is made up of 32 bits of information. These bits are divided into four parts containing 8 bit each.
- IPv4 addresses are unique. Two devices on the internet can never have the same address at the same time.
- Packets in the IPv4 layer are called datagrams. A datagram is a variable length.

b. IPv6

- IPv6 addresses are 128 bits in length. Addresses are assigned to individual interface on nodes, not to the node themselves.
- A single interface may have multiple unique unicast addresses. The first field of any IPv6 address is the variable length format prefix, which identifies various categories of addresses.

c. 6LoWPAN

- IPv6 over Low power Wireless Personal Area Network enables IPv6 in low-power and lossy wireless networks such as WSNs.
- 6LoWPAN defines header compression mechanisms.

3. Transport Layer

- A transport layer protocol provides for logical communication between application processes running on different hosts.
- The transport layer is responsible for delivery of message from one process to another. The network does the host to destination delivery of individual packets considering it as independent packet.

Scanned by TapScanner

- But transport layer ensures that the whole message arrives intact and in order with error control and process control.
- A transport protocol can offer reliable data transfer service to an application even when the underlying network protocol is unreliable, even when the network protocol loses, garbles and duplicate packets.

a. TCP (Transmission Control Protocol)

- TCP is the connection oriented protocol whereas UDP is connectionless protocol. Both are internet protocols used in the transport layer.
- TCP provides a connection-oriented, reliable, byte stream service. The term connection oriented means the two applications using TCP must establish a TCP connection with each other before they can exchange data.
- TCP does not support multicasting and broadcasting. The application data is broken into what TCP considers the best sized chunks to send. The unit of information passed by TCP to IP is called a segment.
- When TCP sends a segment it maintains a timer, waiting for the other end to acknowledge reception of segment. If an acknowledgement isn't received in time, the segment is retransmitted.

b. (UDP) User Datagram Protocol

- UDP is a simple, datagram-oriented, transport layer protocol. This protocol is used in place of TCP.
- UDP is connectionless protocol provides no reliability or flow control mechanisms. It also has no error recovery procedures.
- UDP makes use of the port concept to direct the datagrams to the proper upper-layer applications. UDP serves as a simple application interface to the IP.
- UDP uses port numbers as the addressing mechanism in the transport layer.

4. Application Layer

- Application layer is responsible for accessing the network by user. It provides user interfaces and other supporting services such as e-mail, remote file access, file transfer, sharing database, message handling (X.400), directory services (X.500).

a. HTTP (HyperText Transport Protocol)

- HTTP is an application protocol. HTTP is used to retrieve Web pages from remote servers.
- HTTP uses the services of TCP. HTTP is a stateless protocol. The client initializes the transaction by sending a request message. The server replies by sending a response.

Scanned by TapScanner

- HTTP includes commands such as GET, PUT, POST, HEAD, DELETE, MOVE, LINK and UNLINK.
- HTTP messages are two types : Request and Response
- URL is a standard for specifying any kind of information on the internet. HTTP uses URL.

b. CoAP - Constrained Application Protocol

- CoAP is a specialized web transfer protocol for use with constrained nodes and constrained (e.g., low-power, lossy) networks.
- CoAP is designed for simplicity, low overhead and multicast support in resource-constrained environments.
- CoAP is a web protocol that runs over the UDP for IoT. Datagram Transport Layer Security (DTLS) is used to protect CoAP transmission.
- The protocol is designed for machine-to-machine (M2M) applications such as smart energy and building automation.
- CoAP provides a request/response interaction model between application endpoints, supports built-in discovery of services and resources, and includes key concepts of the Web such as URIs and Internet media types.
- CoAP is designed to easily interface with HTTP for integration with the Web while meeting specialized requirements such as multicast support, very low overhead, and simplicity for constrained environments.

c. Websocket

- WebSocket is a communications protocol, providing full-duplex communication channels over a single TCP connection. The WebSockets protocol does not run over HTTP, instead it is a separate implementation on top of TCP.
- The WebSocket protocol starts by a handshake in order to start the communication and exchange messages formatted as frames.
- After connection is established, messages can be transmitted, either client or server initiated. This means that you can make a dynamic web page where changes occur in real time.
- In that way Websockest communication presents a suitable protocol for IoT world where changes are usually asynchronously occurring and number of clients can be very large.

d. MQTT (Message Queue Telemetry Transport)

- MQTT is Open Connectivity for Mobile, M2M and IoT. MQTT is designed for high latency, low-bandwidth or unreliable networks. The design principle minimizes the network bandwidth and device resource requirements.

Scanned by TapScanner

- MQTT is a lightweight broker-based publish/subscribe messaging protocol designed to be open, simple, lightweight and easy to implement.
- The MQTT protocol works by exchanging a series of MQTT control packets in a defined way. Each control packet has a specific purpose and every bit in the packet is carefully crafted to reduce the data transmitted over the network.
- A MQTT topology has a MQTT server and a MQTT client. MQTT control packet headers are kept as small as possible.
- Having a small header overhead makes this protocol appropriate for IoT by lowering the amount of data transmitted over constrained networks.
- MQTT is the protocol built for M2M and IoT which is used to provide new and revolutionary performance.

e. XMPP (Extensible Messaging Presence Protocol)

- The XMPP is targeted at delivering instant messages and presence information. It is an open and XML - Based protocol.
- Instant messaging (IM) is a service, where communicating parties typically end users send messages in one- to-one or one -to -many fashion in near real - time.
- An open technology for real-time communication, which powers a wide range of applications including instant messaging, presence, multi-party chat, voice and video calls, collaboration, lightweight middleware, content syndication, and generalized routing of XML data.
- XMPP support server-to-server communication and client-to-server communication.

f. DDS (Data Distribution Service)

- The first open international middleware standard directly addressing publish-subscribe communications for real-time and embedded systems.
- The DDS is an Object Management Group (OMG) standard for Pub/Sub that addresses the needs of mission and business critical applications, such as, financial trading, air traffic control and management, and complex supervisory and telemetry systems.
- DDS provides a shared "global data space" where any application can publish the data it has & subscribe to the data it needs.
- DDS is highly configurable by means of QoS settings. Heterogeneous systems can be easily accommodated.

g. AMQP (Advanced Message Queuing Protocol)

- A protocol to communicate between clients and messaging middleware servers (brokers). The Broker is the AMQP Server.

Scanned by TapScanner

- AMQP supports both publish-subscribe model and point-to-point communication, routing and queuing.
- AMQP divides the brokering task between exchanges and message queues, where the first is a router that accepts incoming messages and decides which queues to route the messages to, and the message queue stores messages and sends them to message consumers.
- AMQP supports username and password authentication as well as SASL authorization. It also supports TLS encryption.

1.5 Logical Design of IoT

- It is an abstract representation of an entities and processes but it can not specify low level implementation.

1.5.1 IoT Functional Blocks

- The functional model (FM) is derived from internal and external requirements. Functional view is derived from the Functional Model in conjunction with high-level requirements.
- IoT Functional model identifies Functional Groups (FGs) that is, groups of functionalities, grounded in key concepts of the IoT Domain Model.
- Functional Model is an abstract framework for understanding the main Functionality Groups (FG) and their interactions. This framework defines the common semantics of the main functionalities and will be used for the development of IoT-A compliant Functional Views.
- The Functional Model is not directly tied to a certain technology, application domain, or implementation. It does not explain what the different Functional Components are that make up a certain Functionality Group.
- Fig. 1.5.1 shows IoT functional model.
- The Application, Virtual Entity, IoT Service, and Device FGs are generated by starting from the User, Virtual Entity, Resource, Service, and Device classes from the IoT Domain Model.
- Device functional group contains all the possible functionality hosted by the physical devices. Device functionality includes sensing, actuation, processing, storage, and identification components, the sophistication of which depends on the device capabilities.
- Communication functional group support all the communication used by devices. It uses wired and wireless technology.

Scanned by TapScanner

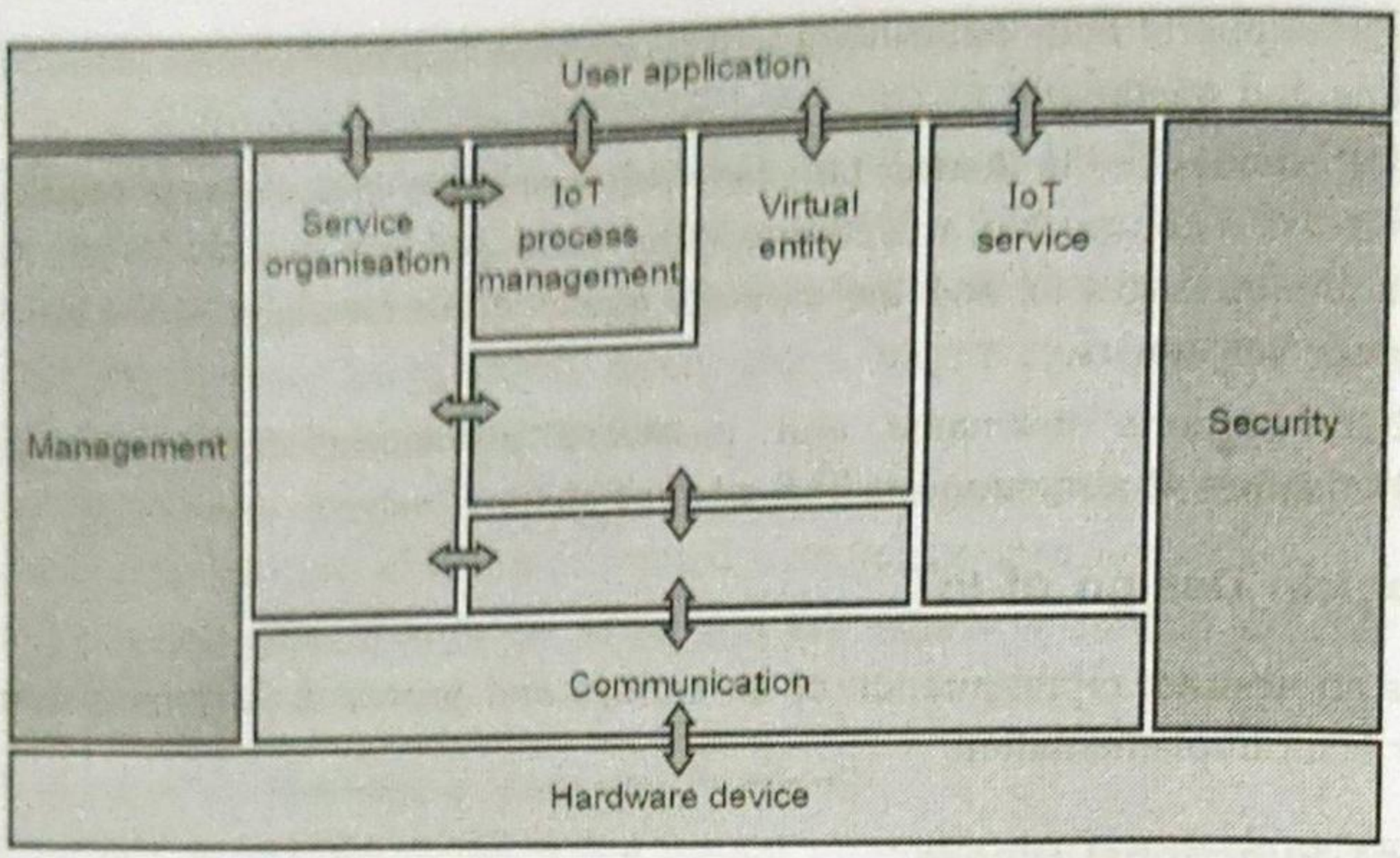


Fig. 1.5.1 : IoT functional model

- IoT Service functional group : Support functions such as directory services, which allow discovery of Services and resolution to resources.
- Virtual Entity functional group : It is related to the Virtual Entity class in the IoT Domain Model. Associations between Virtual Entities can be static or dynamic depending on the mobility of the Physical Entities related to the corresponding Virtual Entities.
- IoT Service Organization functional group : To host all functional components that support the composition and orchestration of IoT and Virtual Entity services.
- Finally, the "Management" transversal FG is required for the management of and/or interaction between the functionality groups.
- The IoT Process Management FG relates to the conceptual integration of (business) process management systems with the IoT ARM.
- The Service Organisation FG is a central Functionality Group that acts as a communication hub between several other Functionality Groups.
- The Virtual Entity and IoT Service FGs include functions that relate to interactions on the Virtual Entity and IoT Service abstraction levels, respectively.
- The Virtual Entity FG contains functions for interacting with the IoT System on the basis of VEs, as well as functionalities for discovering and looking up Services that can provide information about VEs, or which allow the interaction with VEs.
- Communication FG provides a simple interface for instantiating and for managing high-level information flow. It can be customized according to the different requirements defined in the Unified Requirements list.

Scanned by TapScanner

- The Management FG combines all functionalities that are needed to govern an IoT system. The need for management can be traced back to at least four high-level system goals : Cost reduction; Attending unexpected usage issues; Fault handling and Flexibility.
- The Security Functionality Group is responsible for ensuring the security and privacy of IoT-A-compliant systems. It is in charge of handling the initial registration of a client to the system in a secure manner. This ensures that only legitimate clients may access services provided by the IoT infrastructure.

1.5.2 IoT Communication Model

Request/Response model :

- In the Request/Response model, client requests information from the server and waits till the response is served from the server. Fig. 1.5.2 shows Request/Response model.

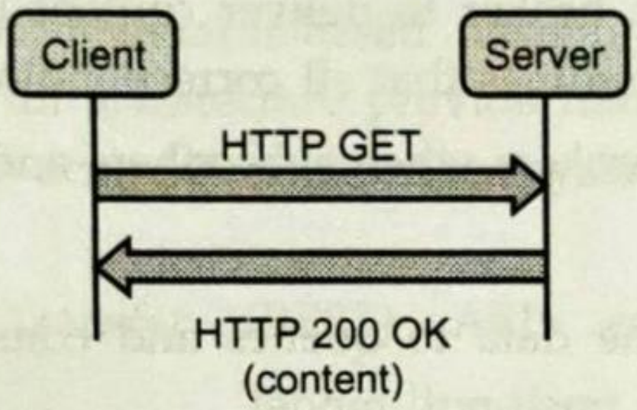


Fig. 1.5.2 Request/Response model

- HTTP protocol is used by Request/Response model. For example, a browser client may request a web page from the server through a "Request" and the corresponding web page will be served by the server as a "Response".
- The client and the server can communicate one to one, or one to many with more requests.
- This model is stateless communication model and each request-response pair is independent of others.

Publish/Subscribe Model:

- Publishers : Publishers generate event data and publishes them.
- Subscribers : Subscribers submit their subscriptions and process the events received
- Publish/Subscribe service : It's the mediator/broker that filters and routes events from publishers to interested subscribers.
- Fig. 1.5.3 shows Publish/Subscribe Model.
- The publishers and subscribers are autonomous, which means that they do not need to know the presence of each other.

Scanned by TapScanner

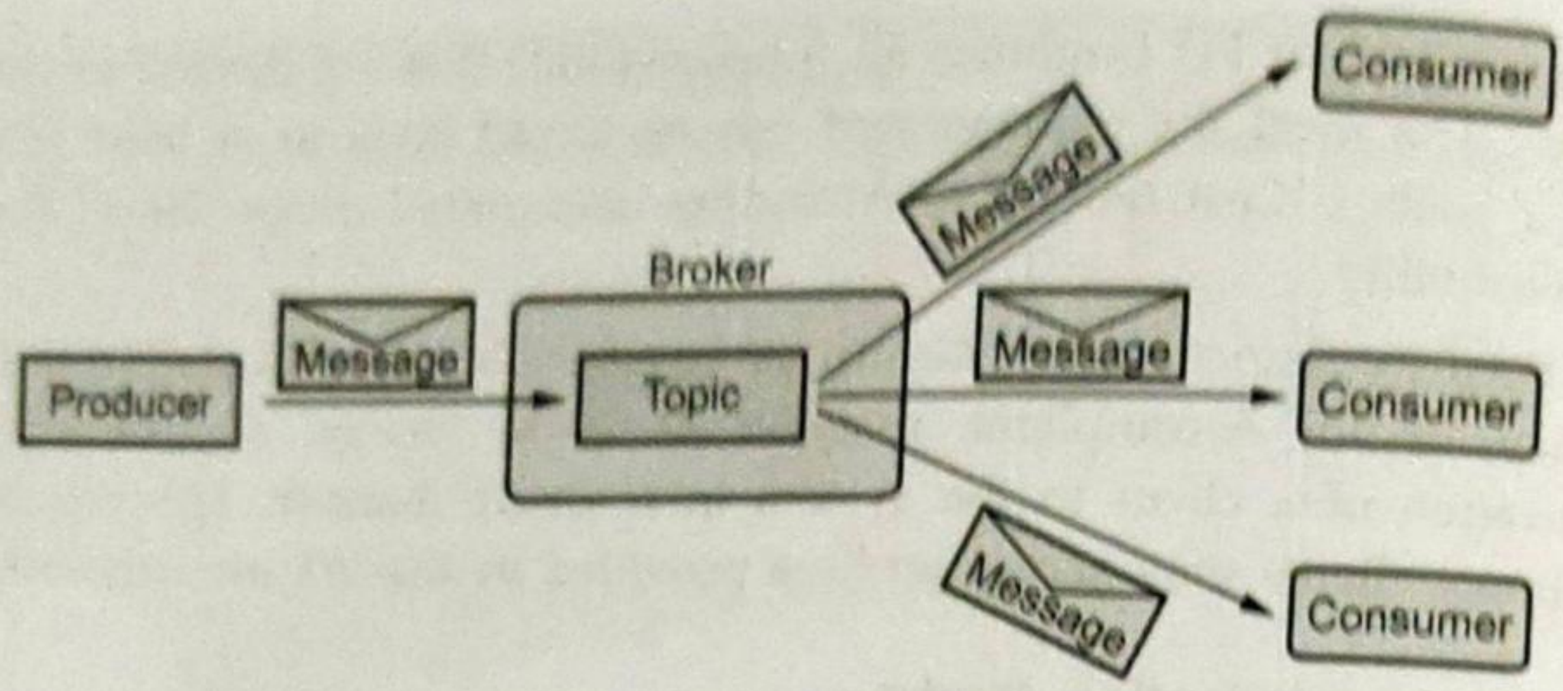


Fig. 1.5.3 Publish/Subscribe model

- This model is highly suited for mobile applications, ubiquitous computing and distributed embedded systems.
- Failure of publishers or subscribers does not bring down the entire system.
- No strong guarantee on broker to deliver content to subscriber. After a publisher publishes the event, it assumes that all corresponding subscribers would receive it.
- Potential bottleneck in brokers when subscribers and publishers overload them.

Push/Pull Model:

- Data procedure push the data to queues and consumers pull the data from the queues. Fig. 1.5.4 shows push-pull model.

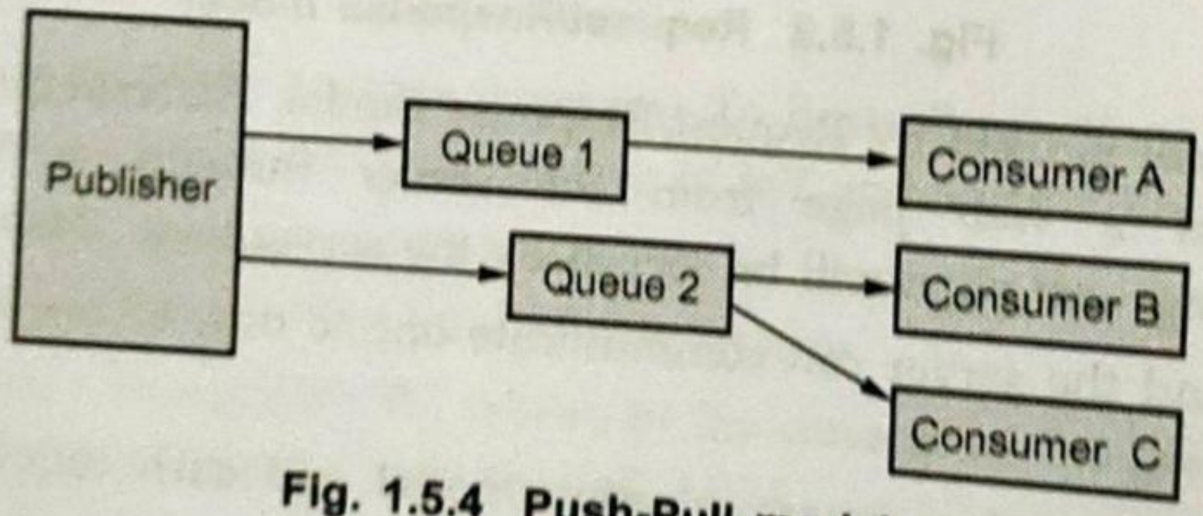


Fig. 1.5.4 Push-Pull model

- Sometimes queue act as buffer in between producer and consumer. Producer does not need to be aware of the consumers.

Exclusive Pair Model:

- This communication model is full duplex, bi-directional communication model. It uses persistent connection between client and server. Fig. 1.5.5 shows exclusive pair model.
- Client send request to server for opening the connection. This connection is open till the client send request for closing the connection.

Scanned by TapScanner

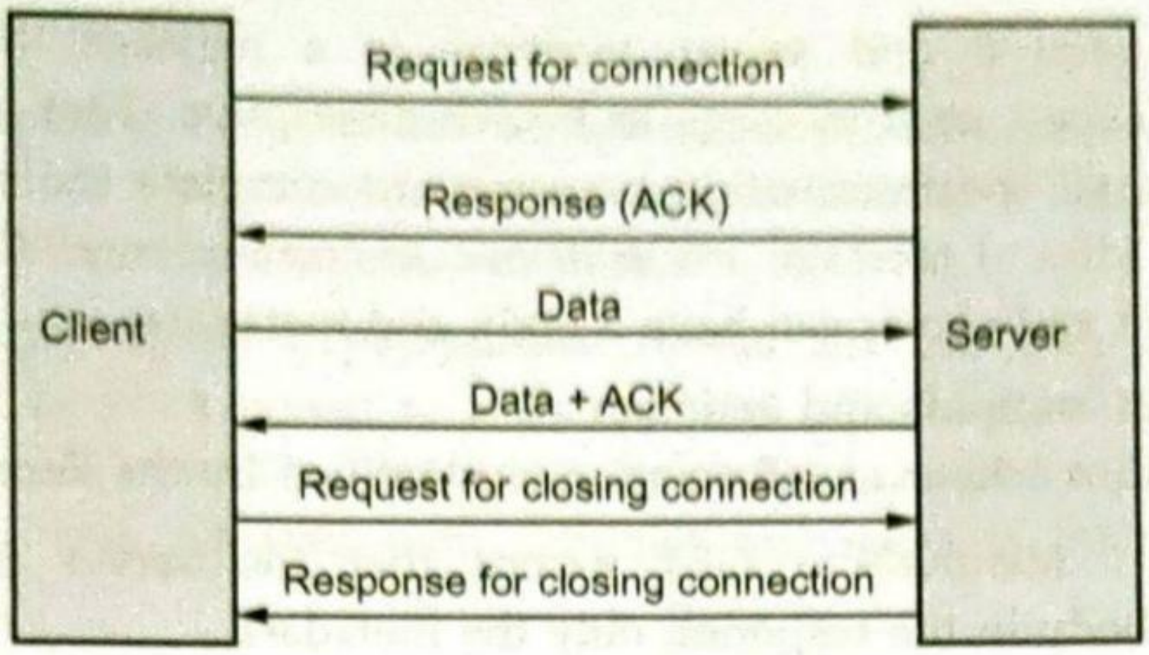


Fig. 1.5.5 Exclusive pair model

1.5.3 IoT Communication API's

1. REST Based Communication API :

- A large part of the interoperability, scale, and control for IoT can be achieved through API management. Standards-based design patterns for Web APIs, API management, and a RESTful architecture provide tremendous value in simplifying the task of interoperability across heterogeneous systems handling vast amounts of data.
- Representational State Transfer (REST) APIs follow the request-response communication model.
- Applications conforming to the REST constraints can be called RESTful. RESTful systems typically communicate over HTTP with the same Methods (GET, POST, PUT, DELETE etc) that browsers use to retrieve web pages and to send data to remote servers.
  1. Client-Server : requires that a service offer one or more operations and that services wait for clients to request these operations.
  2. Stateless : requires communication between service consumer (client) and service provider (server) to be stateless.
  3. Cache : requires responses to be clearly labeled as cacheable or non-cacheable.
  4. Uniform Interface : requires all service providers and consumers within a REST-compliant architecture to share a single common interface for all operations.
  5. Layered System : requires the ability to add or remove intermediaries at runtime without disrupting the system.
  6. Code-on-Demand : allows logic within clients (such as Web browsers) to be updated independently from server-side logic using executable code shipped from service providers to consumers.

Scanned by TapScanner

- Each client request and server response is a message, and REST-compliant applications expect each message to be self-descriptive. That means each message must contain all the information necessary to complete the task. Other ways to describe this kind of message are "stateless" or "context-free." Each message passed between client and server can have a body and metadata.
- **HTTP request methods and actions :**
  1. GET : return whatever information is identified by the Request-URI
  2. HEAD : identical to GET except that the server must not return a message-body in the response, only the metadata
  3. OPTIONS : return information about the communication options available on the request/response chain identified by the Request-URI
  4. PUT : requests that the enclosed entity be stored under the supplied Request-URI
  5. POST : requests that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI
  6. DELETE : requests that the origin server delete the resource identified by the Request-URI.
- The first three are read-only operations, while the last three are write operations.

2. WebSocket based Communication API

- WebSocket support full-duplex, two-way communication between client and server.
- WebSocket APIs reduce the network traffic and latency as there is no overhead for connection setup and termination requests for each message.
- WebSocket uses a standard HTTP request-response sequence to establish a connection. When the connection is established, the WebSocket API provides a read and write interface for reading and writing data over the established connection in an asynchronous full duplex manner.
- WebSocket also provides an interface for asynchronously closing the connection from either side.

1.6 Overview of IoT Protocols

- IoT is enabled by several technologies including wireless sensor networks, cloud computing, Big data analytics, Embedded Systems, Security Protocols and architectures, communication protocols, web services, Mobile Internet, and Semantic Search engines.

Scanned by TapScanner

1.6.1 Cloud Computing

- Cloud computing has the almost unlimited capacity of storage and processing power which is a more mature technology at least to a certain extent to solve the problem of most of the Internet of things.
- Cloud computing is a pay-per-use model for enabling available, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal management effort or service-provider interaction.
- Cloud storage services may be accessed through a web service API, a cloud storage gateway or through a web-based user interface.
- **Cloud computing services are offered to users in different forms :** Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS).
- **Software as a Service (SaaS) :** Model in which an application is hosted as a service to customers who access it via the Internet. The provider does all the patching and upgrades as well as keeping the infrastructure running. The traditional model of software distribution, in which software is purchased for and installed on personal computers.
- **Platform as a Service (PaaS) :** Platform as a service is another application delivery model and also known as cloud-ware. Supplies all the resources required to build applications and services completely from the Internet, without having to download or install software. Services includes application design, development, testing, deployment, and hosting, team collaboration, web service integration, database integration, security, scalability, storage, state management, and versioning. PaaS is closely related to SaaS but delivers a platform from which to work rather than an application to work with.
- **Infrastructure as a Service (IaaS) :** SaaS and PaaS are providing apples to customers, IaaS doesn't. It offers the hardware so that your organization can put whatever they want onto it. Rather than purchase servers, software, racks, and having to pay for the datacenter space for them, the service provider rents for resources like server space, network equipment, memory etc.

1.6.2 Big Data Analytic

- A category of technologies and services where the capabilities provided to collect, store, search, share, analyze and visualize data which have the characteristics of high-volume, high-velocity and high-variety.

Scanned by TapScanner

- Examples of big data generated by IoT systems :
  a) Weather Monitoring Stations
  b) Machine sensor data from industrial systems
  c) Health and fitness data
  d) Location and tracking systems

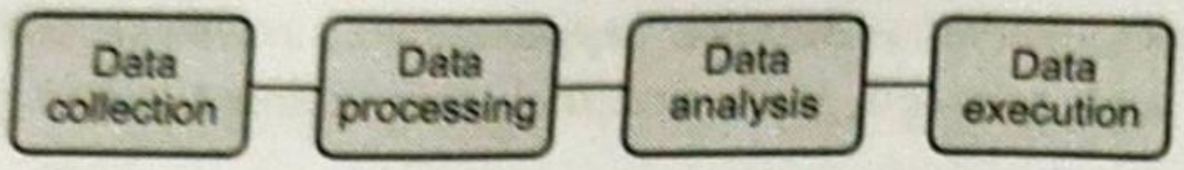


Fig. 1.6.1

1.6.3 Wireless Sensor Networks

- A Wireless Sensor Network (WSN) is a network formed by a large number of sensor nodes where each node is equipped with a sensor to detect physical phenomena such as light, heat, pressure, etc.
- WSNs nowadays usually include sensor nodes, actuator nodes, gateways and clients. A large number of sensor nodes deployed randomly inside of or near the monitoring area, form networks through self-organization.
- Sensor nodes monitor the collected data to transmit along to other sensor nodes by hopping. During the process of transmission, monitored data may be handled by multiple nodes to get to gateway node after multi-hop routing, and finally reach the management node through the internet or satellite.
- Standards for WSN technology have been well developed, such as Zigbee (IEEE 802.15.4). The IEEE 802.15.4 is simple packet data protocol for lightweight wireless networks.
- It works well for long battery life, selectable latency for controllers, sensors, remote monitoring and portable electronics.

1.6.4 Communication Protocols

- Communication protocols are used as a backbone of IoT systems. It enable network connectivity and coupling to applications. Communication protocols allow devices to exchange data over the network.
- Communication protocol also performs error correction and detection, flow control, data encoding, addressing mechanism etc.
- Sequence control, lost of packet, retransmission are the other functions of communication protocol.

Scanned by TapScanner

1.6.5 Embedded System

- A system is a set of interacting or interdependent component parts forming a complex unit. It is a fixed plan to perform one or many task.
- Embedded system is an electronic system which is designed to perform one or a limited set of functions using software and hardware.
- General definition of embedded systems is : embedded systems are computing systems with tightly coupled hardware and software integration that are designed to perform a dedicated function. The word embedded reflects the fact that these systems are usually an integral part of a larger system, known as the embedding system. Multiple embedded systems can coexist in an embedding system.
- An embedded system has three main components: Hardware, Software and time operating system.
- Hardware parts includes power supply, processor, memory, times & communication ports, system application specific circuit etc.
- Software parts includes the application software is required to perform the series of tasks. An embedded system has software designed to keep in view of three constraints :
  a. Availability of System Memory
  b. Availability of processor speed
  c. The need to limit power dissipation when running the system continuously in cycles of wait for events, run , stop and wake up.
- Demand for low cost and higher density platform requires the integration of devices. As integration levels increases, more and more logic is added to the processor die, creating families of applications specific service processors.
- System-on-Chip (SoC) designs increasingly become the driving force of a number of modern electronics systems. A number of key technologies integrate together in forming the highly complex embedded platform.

1.7 IoT Levels and Deployment Templates

- IoT system consists of following components
  1 Device : IoT device performs identification, remote monitoring, sensing and actuating functions.
  2 Resource : IoT device used software components as resources for accessing, processing, and storing sensor information. It also controls actuators.
  3 Controller Service : It sends data from the device to the web service and receives commands from the application for controlling the device.
  4 Database : IoT generates large amount of local database and cloud database.

Scanned by TapScanner

5 Web Service : Web services act as a link between the IoT device, application, database and analysis components.

6 Analysis Component : It is responsible for analysing the IoT data and generate results.

7. Application : User use this interface for controlling and monitoring various IoT system.

- Levels of IoT system are IoT Level-1, IoT Level-2, IoT Level-3, IoT Level-4, IoT Level-5, IoT Level-6.
- IoT Development level are six types.

IoT Level 1	Single node, perform sensing, perform analysis and hosts the application
IoT Level 2	Single node, perform sensing, perform local analysis
IoT Level 3	Single node, data is analyzed in cloud and application is cloud based
IoT Level 4	Multiple Node, perform local analysis, application is cloud based
IoT Level 5	Multiple end node and coordinator node,
IoT Level 6	Multiple independent node, perform sensing, send data to cloud

1.7.1 IoT Level 1

- Physical devices and controllers that might control multiple devices. These are the "things" in the IoT, and they include a wide range of endpoint devices that send and receive information.
- Fig. 1.7.1 shows IoT level 1.

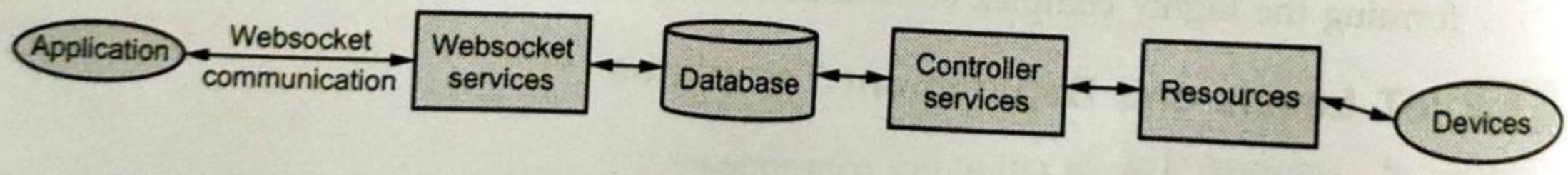


Fig. 1.7.1 IoT level 1

- Level 1 IoT systems are suitable for modeling low-cost and low complexity solutions where the data involved is not big and the analysis requirements are not computationally intensive.
- Eg : Home automation
- The system consists of a single node that allows controlling the lights and appliances in a home remotely. The device used in the system interfaces with the

Scanned by TapScanner

light and appliances using electronic relay switches. The status information of each light or appliance is maintained in the local database. The controller service continuously monitors the state of each light or appliance and triggers the relay switches accordingly.

1.7.2 IoT Level 2

- In level 2, single node performs sensing and local analysis. Fig. 1.7.2 shows block diagram.

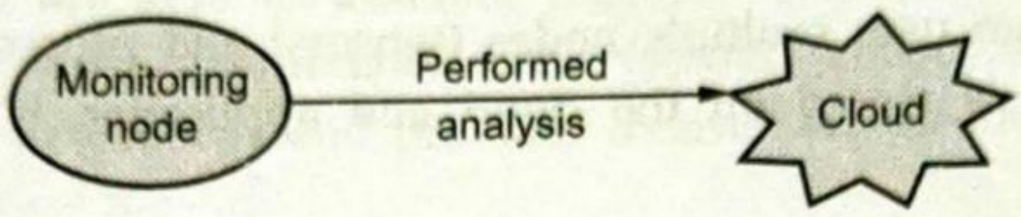


Fig. 1.7.2

- Both data and application is stored on the cloud. This type of system is suitable for big data used for analysis and that to performed on local side.
- Smart irrigation is an example of level 2 IoT system. System uses single sensor for monitoring the soil moisture level and controls the irrigation system.
- Controller service monitors continuously the moisture level. If the moisture level drops below a threshold, the irrigation system is turned ON.

1.7.3 IoT Level 3

- Single node is used in level 3 internet of things. Collected data is stored on the cloud and processed on the cloud. Application is also stored on the cloud.
- This is suitable for huge data and analysis requirement are computationally intensive.

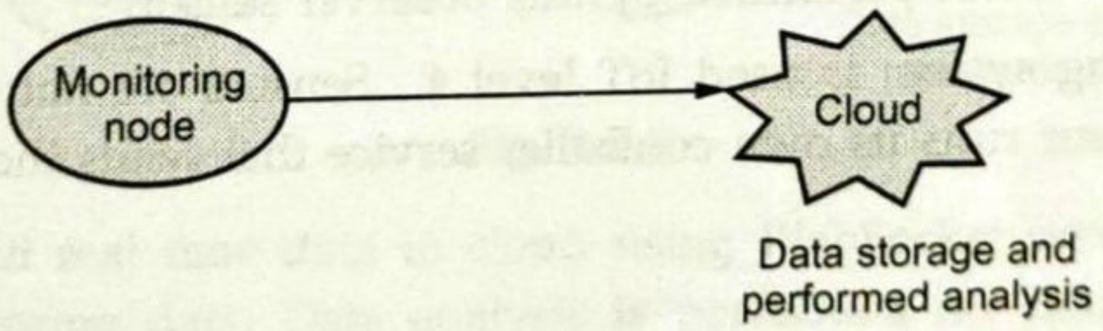


Fig. 1.7.3

- REST or websocket is used for communicating with client machine. Consider an example of tracking package handling.
- Single node monitors the vibration level for the package being shipped. For monitoring vibration level, accelerometer and gyroscope sensor is used.

Scanned by TapScanner

- Accelerometer and gyroscope sensor is used. System allows shippers tracking cargo to communicate with the devices, changing reporting frequencies and investigating alerts generated by attached sensors.
- Controller device sends the sensor data to the cloud in real time using WebSocket service. Data is stored on the cloud and analysis is also performed on the cloud.
- Cloud can send alerts if the vibration levels is greater than threshold level.

1.7.4 IoT Level 4

- Level 4 IoT system uses multiple nodes (sensors) and processing is performed on local node. Data is stored on the cloud and application is stored on the cloud storage.
- Fig. 1.7.4 shows block diagram of Level 4 IoT.

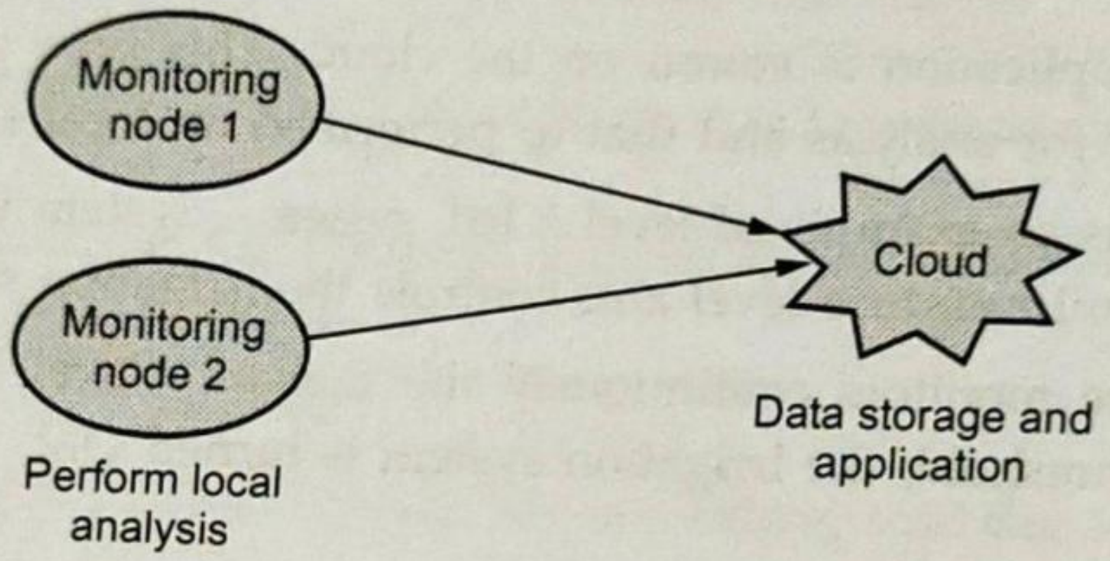


Fig. 1.7.4

- Level 4 IoT uses two observer sensor for local and cloud. These sensor subscribe and receive information form the IoT device to cloud .
- The observer sensor can process information and used for various purposes. Control function is not performed by this observer sensor.
- Noise monitoring system is used IoT level 4. Sensors are not depends upon each other. Each sensor runs its own controller service that sends the data to the cloud.

1.7.5 IoT Level 5

- It contains multiple end sensor and one coordinator sensor. Fig. 1.7.5 shows IoT level 5 block diagram.
- The end sensor perform sensing and/or actuation. Coordinator sensor collect data from the end sensor and sends to the cloud.
- Data is stored and analysis in the cloud and application is also cloud based. Forest fire detection system uses level 5 IoT system.

Scanned by TapScanner

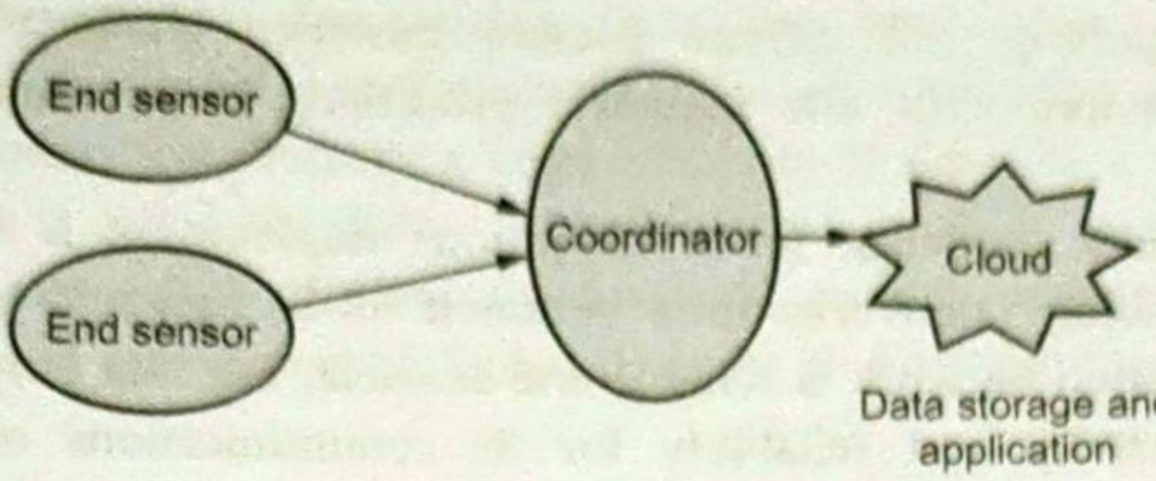


Fig. 1.7.5

- Multiple sensor are kept at different location to monitor temperature, humidity, carbon dioxide level. Coordinator node (sensor) collect data for end node and these nodes act as gateway and provides Internet services to the IoT systems.

1.7.6 IoT Level 6

- It contains multiple independent end nodes and it performs sensing and/or actuation function. It sends data to the cloud. Fig. 1.7.6 shows block diagram of level 6 IoT.
- Data is stored on the cloud and application is also cloud based. Result is displayed on the cloud. Centralized controller knows the status of monitoring node and sends control commands to the nodes.
- Weather monitoring system uses Level 2 IoT based system. Multiple nodes are kept at different location to monitor temperature, humidity level.

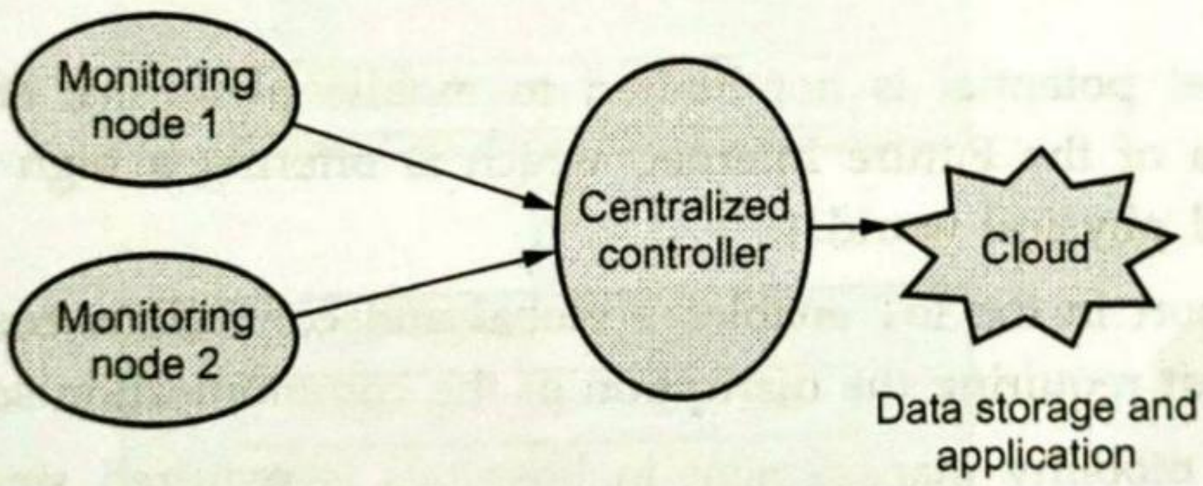


Fig. 1.7.6

- End node send real time data to cloud using WebSocket service. Cloud database is used for storing data. Data analysis is performed on cloud side. Cloud based application is used for display data.

1.8 Challenges for IoT

- IoT and ubiquitous integration of clinical environments define complex design challenges and requirements in order to reach a suitable technology maturity for its wide deployment and market integration.

Scanned by TapScanner

- From the beginning, IoT devices present inherited challenges since they are constrained devices with low memory, processing, communication and energy capabilities.
  1. The first key challenge for a ubiquitous deployment is the integration of multi-technology networks in a common all-IP network to ensure that the communication network is reliable and scalable. For this purpose, IoT relies on the connectivity and reliability for its communications on Future Internet architecture.
  2. The second key challenge is to guarantee security, privacy, integrity of information and user confidentiality. The majority of the IoT applications need to take into considerations the support of mechanisms to carry out the authentication, authorization, access control, and key management.
  3. In addition, due to the reduced capabilities from the constrained devices enabled with Internet connectivity, a higher protection of the edge networks needs to be considered with respect to the global network.
  4. The third key challenge is to offer support for the mobility, since the Future Internet presents a more ubiquitous and mobile Internet. Mobility support increases the applicability of Internet to new areas.
- The most present nowadays are mobile platforms such as smart phones and tablets which enable a tremendous range of applications based on ubiquitous location, context awareness, social networking, and interaction with the environment.
- Future Internet potential is not limited to mobile platforms, else IoT is another emerging area of the Future Internet, which is offering a high integration of the cybernetic and physical world.
- Mobility support in the IoT enables a global and continuous connection of all the devices without requiring the disruption of the communication sessions.
- For example, mobility management in hospitals is required since clinical devices can be connected through wireless technologies. Mobility offers highly valuable features such as higher quality of experiences for the patients, since this allows the patients to move freely, continuous monitoring through portable/wearable sensors, extend the coverage within all the hospital, and finally a higher fault tolerance since the mobility management allows the connection to adapt dynamically to different access points.
- Therefore, clinical environment is one of the main scenarios where the mobility for the IoT-based applications exploit these capabilities, in terms of fault tolerance influences directly in the life support, and continuous monitoring influences the quantity of data available which is required for real-time diagnostic.

Scanned by TapScanner

5. Other challenges are also arising from the application, economical, and technological perspectives. For example, from an application point of view are the requirements for processing large amounts of data for a growing number of devices, it is called Big Data.

- From the economic points of view, the needs to provide economies of scale, i.e., new services based on existing modules in order to leverage the related platform investment.
- From the networking point of view to offer an end-to-end support for Quality of Service , since the different IoT applications will present different requirements in terms of latency and bandwidth, for example, for clinical environments the traffic should be prioritized over other non-critical traffic coming from smart-metering.
- Fig. 1.8.1 shows the key challenges to offer an Internet of Everything. This covers from the integration of heterogeneous devices to the integration into a Web of Things.

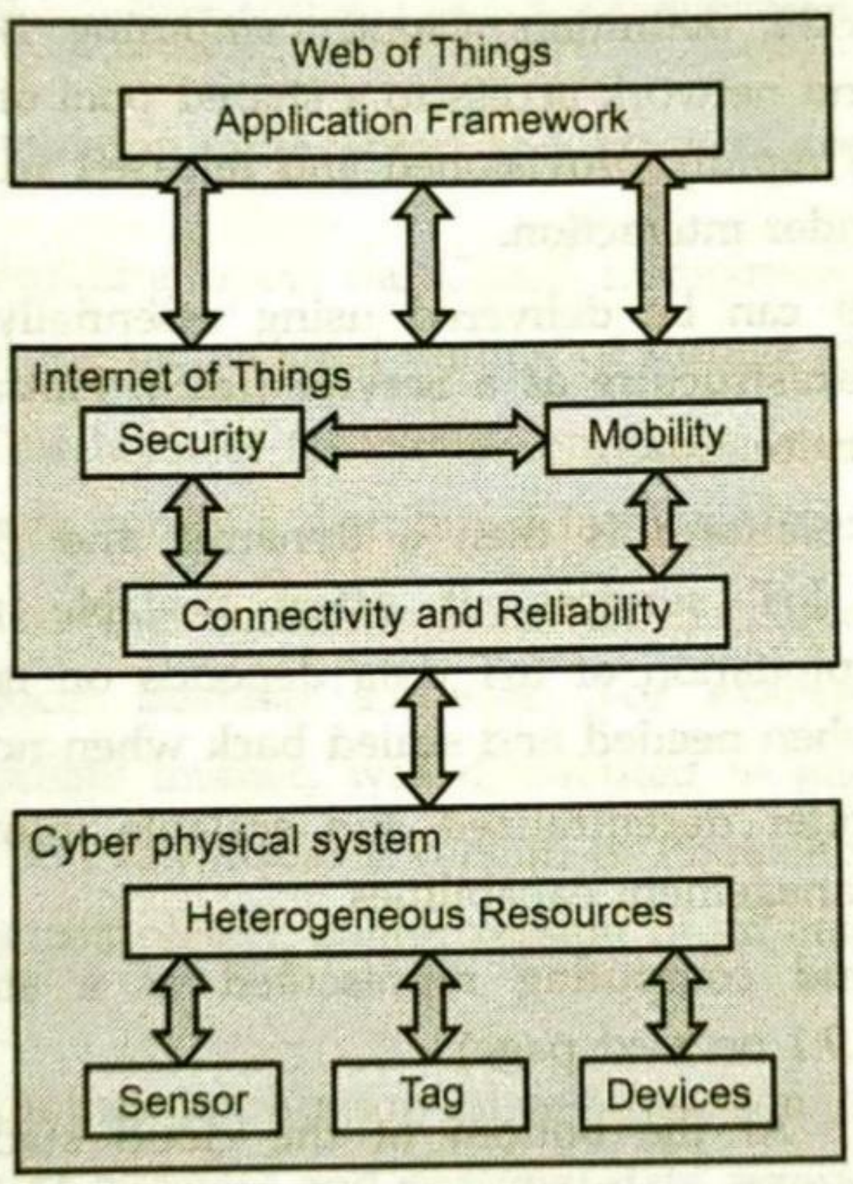


Fig. 1.8.1 Key challenges to offer an Internet of Everything

- The number of devices that are connected to the Internet is growing exponentially. This has led to defining a new conception of Internet, the commonly called Future Internet, which started with a new version of the Internet Protocol (IPv6) that extends the addressing space in order to support all the emerging Internet-enabled devices.

Scanned by TapScanner

- IoT devices are small wireless devices that would be placed in public places. Wireless communication is made secure through encryption technique. But the IoT devices are very small and not powerful enough to support encryption methods. There is need to modify encryption algorithm in order to support IoT devices.
- In IoT, the different devices are traceable through the interconnected network, it creates threats to personal and private data.
- Many IoT devices are small in size and do not have the continuous power source. A device computation depends on battery size and cost of the device. Many IoT devices work as a single, limited purpose which could have customized network interfaces, operating systems, and programming models that make the most efficient use of limited computation, network, and energy resources.

1.9 Interdependencies of IoT and Cloud Computing

- According to the NIST definition, Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction.
- The Cloud paradigm can be delivered using essentially three different service models. These are Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).
- A Cloud-based IoT platform is then a dynamic and flexible resource sharing platform delivering IoT services. It offers scalable resources and services management. The exploitation of IoT data depends on massive resources, which should be available when needed and scaled back when not needed.
- Cloud technologies offer decentralized and scalable information processing and analytics, and data management capabilities.
- Fig. 1.9.1 shows cloud computing represented as a stack of service offering categories. (See Fig. 1.9.1 on next page)
- Cloud Infrastructure : At the bottom of the cloud stack, Cloud Infrastructure provides the distributed multi-site physical components to support cloud computing, such as storage and processing resources. This layer allows the infrastructure provider to abstract away details such as which exact hardware an application is using and which data center the application is running in.
- Cloud Platform : Platform offerings provide an infrastructure for developing and operating web-based software applications. Examples include facilities for application design, application development, testing, deployment, and hosting, as well as application services such as team collaboration, security, application versioning, and application instrumentation.

Scanned by TapScanner

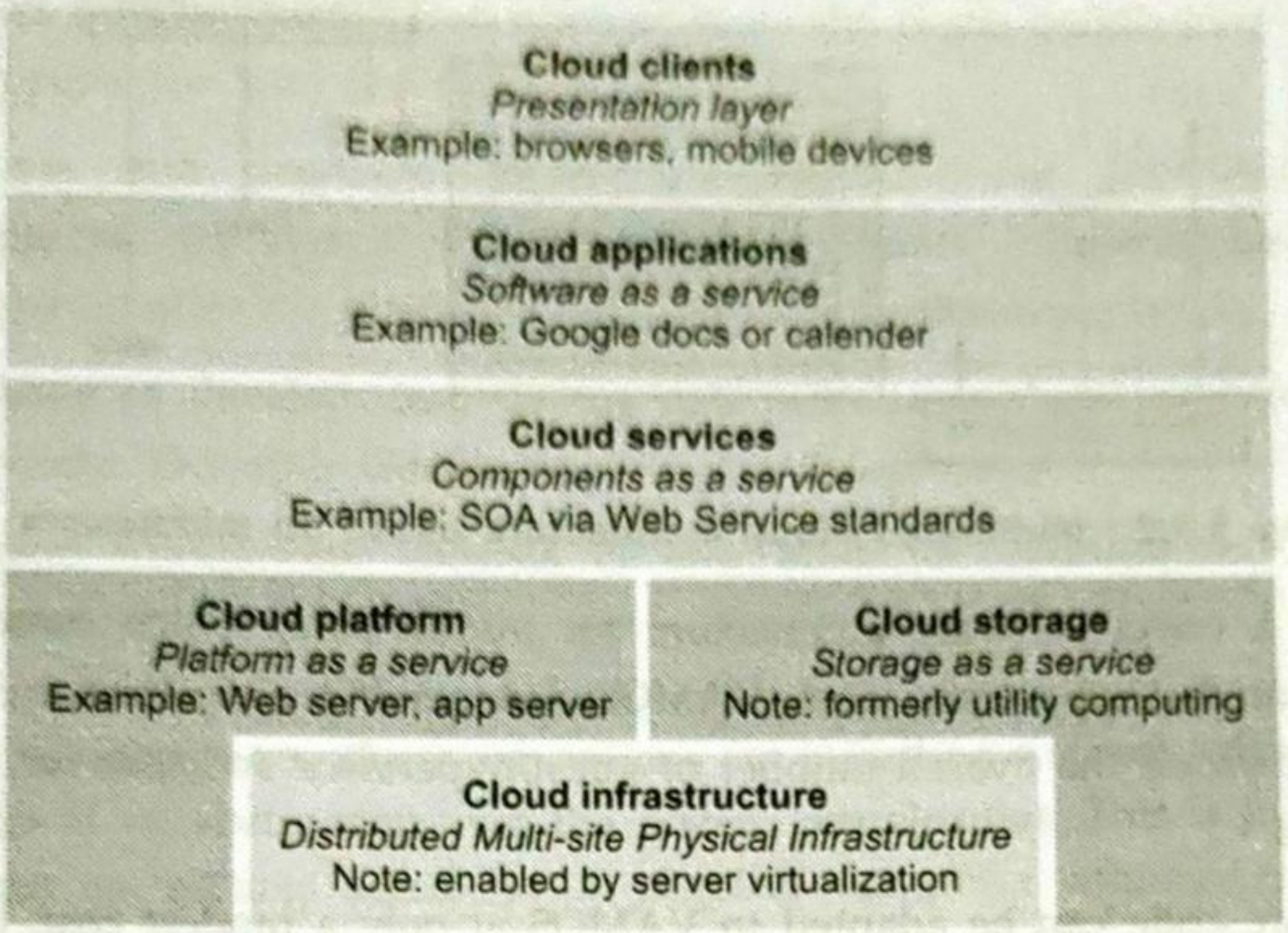


Fig. 1.9.1 : Cloud computing represented as a stack of service offering catagories

- Cloud Storage : Building upon the Cloud Infrastructure, this layer of the cloud stack is focused on the incremental renting of storage on the Internet.
- Cloud Services : This layer of the cloud computing stack includes the definition of software components, run in a distributed fashion, across the commercial Internet.
- Cloud Applications : This definition relies on the cloud for access to what would traditionally be local desktop software. For example, Adobe's Photoshop, a program to manipulate images, was distributed to end users on disks for many years. Today, you still can install a version of Photoshop from an installation disk, or you can go to a completely online version of an analogous application, entitled Express.
- Cloud Clients : Another application-related function of cloud computing focuses on the distribution of business and personal data across servers on the Internet.

1.9.1 Cloud Middleware

- Middleware helps in reducing the overhead of virtualization. Fig. 1.9.2 shows Multitiered cloud architecture based on middleware.

Scanned by TapScanner

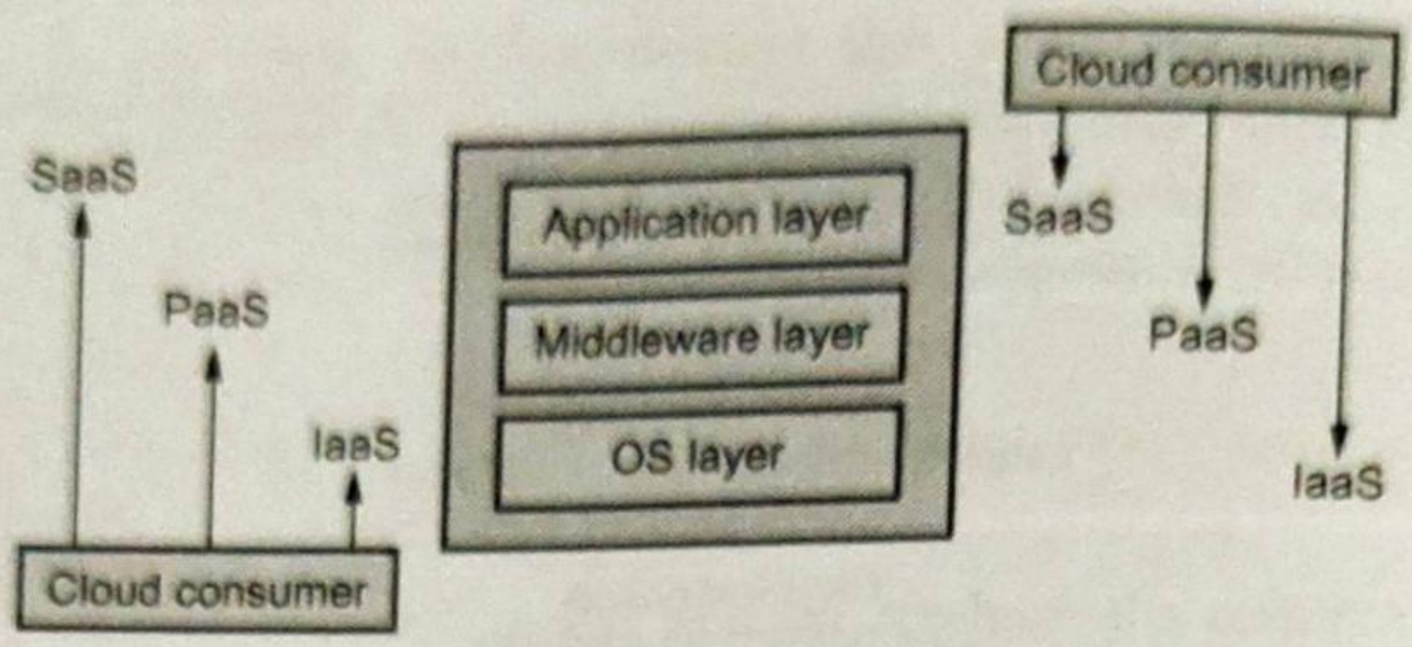


Fig. 1.9.2 : Multitiered cloud architecture based on middleware

- VAMOS, a novel software architecture for middleware, which runs middleware modules at the hypervisor level. VAMOS reduces I/O virtualization overhead by cutting down on the overall number of guest/hypervisor switches for I/O intensive workloads.
- Middleware code can be adapted to VAMOS at only a modest cost, by exploiting existing modular design and abstraction layers. Applying VAMOS to a database workload improved its performance by up to 32 %.
- VAMOS does not require changes in the guest OS and take advantage of the host OS by running selected middleware modules directly at the hypervisor level, minimizing the adaption cost and re-using existing code.
- The parallel computing environments such as PVM and MPI are (HPC) middleware by definition. The HPC middleware fills the gap that the operating systems and the programming languages lack to support parallel computing.
- Grid computing is the foundation of cloud computing infrastructure, so grid middleware is the basis of IaaS middleware.
- The grid computing middleware software will manage and execute all the activities related to identification, allocation, de-allocation and consolidation of all the computing resources to the end-users transparently, as in the case of a geographical distributed resources system.
- Grid computing environment is a necessity to many end-users who cannot afford huge computational resources, both hardware and software. Therefore, any large corporate body or government organization having a large geographical spread will be essentially required to set up at least some kind of grid computing environment, so that the expensive resources of their grid can be shared and effectively utilized by all the end users.

Scanned by TapScanner

- The PaaS middleware is often referred to as the cloud middleware that underpins and supports the SaaS applications.
- Provisions and manages cloud infrastructure and middleware; provides development, deployment and administration tools. Infrastructure providers can transparently alter the platforms for their customers' unique needs.
- Users have to develop, test, deploy and manage applications hosted in a cloud environment. Example: Google App Engine, SalesForce.
- The cloud middleware consists of two kinds of middleware: IaaS and PaaS middlewar
  1. Infrastructure as a Service (IaaS) : The raw computing resources are exposed to the consumers. There are options for including support software but usually there is no abstraction over the system complexities. This is in fact exactly the same as getting access to a physical computer attached to the Internet. Amazon EC2 is the leading IaaS provider and the most prominent cloud service provider today.
  2. Platform as a Service (PaaS) : Consumers get access to an application development platform possibly hosted on a infrastructure cloud. The platform completely abstracts the complexities of the underlying host system. The platform also guarantees load balancing and scaling in a transparent manner to the cloud consumer. However these platforms typically are restrictive and requires the hosted applications to strictly adhere to certain specialized API, frameworks, and programming languages. Google AppEngine is an example of a PaaS.

1.9.2 Cloud Standards

- Cloud computing standardization organizations
  1. NIST : Working definition of cloud computing
  2. Distributed Management Task Force
  3. Cloud Management Working Group
  4. European Telecommunications Standards Institute
  5. Standards Acceleration to Jumpstart Adoption of Cloud Computing
  6. Open Cloud Consortium

Scanned by TapScanner

- Cloud Standardization Efforts are listed below :

Project Name	URL	Focus
CloudAudit, also known as Automated Audit, Assertion, Assessment, and Assurance API (A6)	http://www.cloudaudit.org	• Open, extensible, and secure interface, namespace, and methodology for cloud-computing providers and their authorized consumers to automate the audit, assertion, assessment, and assurance of their environments • Part of the Cloud Security Alliance since October 2010
Cloud Computing Interoperability Forum	http://www.cloudforum.org	• Common, agreed-on framework/ontology for cloud platforms to exchange information in a unified manner • Sponsors of the Unified Cloud Interface Project to create an open and standardized cloud interface for the unification of various cloud APIs
Cloud Security Alliance	http://cloudsecurityalliance.org	• Recommended practices for cloud-computing security • Working on Version 3 of the Security Guidance for Critical Areas of Focus in Cloud Computing • Nonprofit organization that includes Google, Microsoft, Rackspace, Terre-mark, and others

Scanned by TapScanner

Cloud Standards Customer Council	http://cloudstandardscustomercouncil.org	• Standards, security, and interoperability issues related to migration to the cloud. • End-user advocacy group sponsored by the Object Management Group (OMG) and creator of the Open Cloud Manifesto.
Cloud Storage Initiative	http://www.snia.org/cloud	• Adoption of cloud storage as a new delivery model (Data-Storage-as-a-Service) • Initiative sponsored by the Storage Net-working Industry Association (SNIA), the creator and promoter of the Cloud Data Management Interface (CDMI) • SNIA includes members from NetApp, Oracle, and EMC
DeltaCloud	http://incubator.apache.org/deltacloud	• Abstraction layer for dealing with differences among IaaS providers • API based on representational state transfer (REST) with a small number of operations for managing instances • Currently has libraries for seven providers including Amazon EC2, Eucalyptus, and Rackspace
Distributed Management Task Force (DMTF)	http://dmtf.org/standards/cloud	• Management interoperability for cloud systems • Developer of the Open Virtualization Framework (OVF) • Runs the Open Cloud Standards Incubator

Scanned by TapScanner

IEEE P2301, Guide for Cloud Portability and Interoperability Profiles	http://standards.ieee.org/develop/project/2301.html	• Standards-based options for application interfaces, portability interfaces, management interfaces, interoperability inter-faces, file formats, and operation conventions
IEEE P2302, Draft Standard for Inter-cloud Interoperability and Federation	http://standards.ieee.org/develop/project/2302.html	• Protocols for exchanging data, programmatic queries, functions, and governance for clouds sharing data or functions or for federating one cloud to another
Open Cloud Computing Interface	http://occi-wg.org	• REST-based interfaces for management of cloud resources including computing, storage, and bandwidth • Working group of the Open Grid Forum
Open Cloud Consortium	http://opencloudconsortium.org	• Frameworks for interoperating between clouds and operation of the Open Cloud Testbed

1.9.3 The Cloud of Things Architecture

- Cloud computing creates a new way of designing, developing, testing, deploying running and maintaining applications on the Internet. The Internet of Things and Cloud computing are both emerging technologies and have their own features.
- Fig. 1.9.3 shows the main features of the Cloud-based IoT platform.
- CloudThings architecture is an online platform that allows system integrators and solution providers to leverage a complete. Things application infrastructure for developing, deploying, operating, and composing Things applications and services that consist of three major modules :
  1. The CloudThings service platform for Things is a set of Cloud services (IaaS), allowing users to run any applications on Cloud hardware. The CloudThings service platform for Things dramatically simplifies the application development, eliminates need for infrastructure development, shortens time to market, and reduces Things management and maintenance costs.

Scanned by TapScanner

- The CloudThings service platform offers users unique device management capabilities. It communicates directly with devices and provides storage to collect Things data and transmit Things events. Vast amount of sensor data can be processed, analyzed, and stored using the computational and storage resources of the Cloud.
- The CloudThings service platform allows sharing of sensor resources by different users and applications under a flexible usage mode.

2. The CloudThings Developer Suite for Things is a set of Cloud service tools (PaaS) for Things application development. These tools include open Web service Application Programming Interfaces (APIs), which provide complete development and deployment capabilities to Things developers.

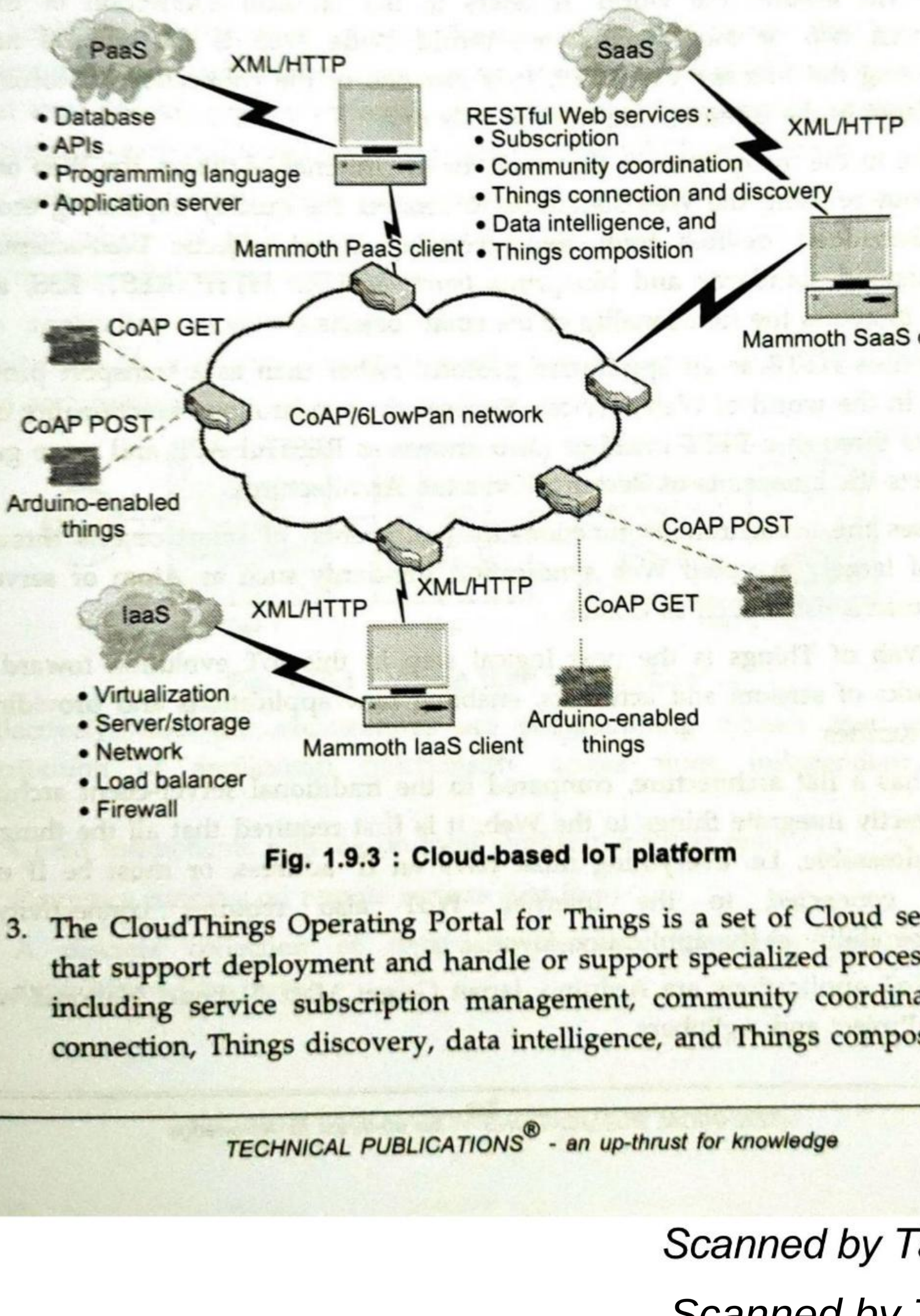


Fig. 1.9.3 : Cloud-based IoT platform

3. The CloudThings Operating Portal for Things is a set of Cloud services (SaaS) that support deployment and handle or support specialized processing services including service subscription management, community coordination, Things connection, Things discovery, data intelligence, and Things composition.

Scanned by TapScanner

1.10 Web of Things

- Web of Things provides an Application Layer that simplifies the creation of Internet of Things applications. The Web, is a system of interlinked documents accessed via the Internet. The Web was originated from Tim Berners-Lee around 1990.
- The Web, like Email, is one of the services that runs on the Internet. Key components of web are as follows :
  1. Uniform Resource Locator (URL) & Uniform Resource Identifier (URI)
  2. HyperText Markup Language (HTML)
  3. Hypertext Transfer Protocol (HTTP)
- Internet is the term used to identify the massive interconnection of computer networks around the world. It refers to the physical connection of the paths between two or more computers. World Wide Web is the general name for accessing the Internet via HTTP. It is just one of the connection protocols that is available in the Internet, and not the only one.
- Unlike in the many systems that exist for the Internet of things, the Web of Things is about re-using the Web standards to connect the quickly expending eco-system of Embedded devices built into everyday smart objects. Well-accepted and understood standards and blueprints (such as URI, HTTP, REST, RSS, etc.) are used to access the functionality of the smart objects.
- WoT uses HTTP as an application protocol rather than as a transport protocol as done in the world of Web Services. Exposes the synchronous functionality of smart objects through a REST interface (also known as RESTful API) and more generally respects the blueprints of Resource Oriented Architectures.
- Exposes the asynchronous functionality (i.e. events) of smart objects through the use of largely accepted Web syndication standards such as Atom or server-push Web mechanisms such as Comet.
- The Web of Things is the next logical step in this IoT evolution toward global networks of sensors and actuators, enabling new applications and providing new opportunities.
- WoT has a flat architecture, compared to the traditional server-client architecture. To directly integrate things to the Web, it is first required that all the things must be addressable, i.e. everything must have an IP address, or must be IP-enabled when connected to the Internet. WoT also requires connectivity and interoperability at the application layer.
- The WoT applications are Arduino, Japan Geiger Map, Nanode, National Weather Study Project and AgSphere.

Scanned by TapScanner

1. Arduino is an open-source electronics platform based on easy-to-use hardware and software.
2. Regulations per the Japanese Nuclear Safety Commission prescribe some standards that a monitoring system at a power producing nuclear plant must adhere to.
3. Nanode is an open- source Arduino- like board that has built-in web connectivity.
4. The National Weather Study Project (NWSP) is a large- scale environmental study project deploying hundreds of mini weather stations in schools throughout Singapore.

1.10.1 Two Pillars of The Web

- An application server is a software framework or middleware that provides an environment in which applications can run, no matter what the applications are or what they do. An application server acts as a set of components accessible to the software developer through an API defined by the middleware itself.
- For web applications, these components are usually performed in the same machine where the web server is running, and their main job is to support the construction of dynamic web pages.
- The application server is based on the three-tiered or multi-tiered software architecture. A "tier" can also be referred to as a "layer". Fig. 1.10.1 shows three tired architecture.

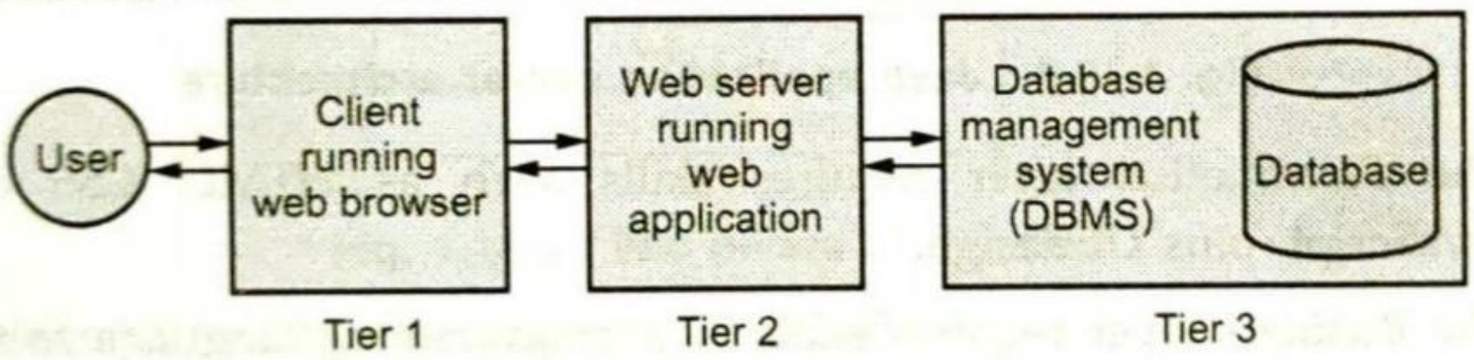


Fig. 1.10.1 : Three tired architecture

- Collectively, three-tier architectures are programming models that enable the distribution of application functionality across three independent systems, typically :
  1. Client components running on local workstations (tier one)
  2. Processes running on remote servers (tier two)
  3. A discrete collection of databases, resource managers, and mainframe applications (tier three)

Scanned by TapScanner

- First tier : Responsibility for presentation and user interaction resides with the first-tier components. These client components enable the user to interact with the second-tier processes in a secure and intuitive manner.
- Second tier : The second-tier processes are commonly referred to as the application logic layer. These processes manage the business logic of the application, and are permitted access to the third-tier services. The application logic layer is where most of the processing work occurs.
- Third tier : The third-tier services are protected from direct access by the client components residing within a secure network. Interaction must occur through the second-tier processes.
- Communication among tiers : All three tiers must communicate with each other. Open, standard protocols and exposed APIs simplify this communication. You can write client components in any programming language, such as Java or C++. These clients run on any operating system, by speaking with the application logic layer. Databases in the third tier can be of any design, if the application layer can query and manipulate them. The key to this architecture is the application logic layer.
- The Java EE standard - based application server architecture is shown Fig. 1.10.2.

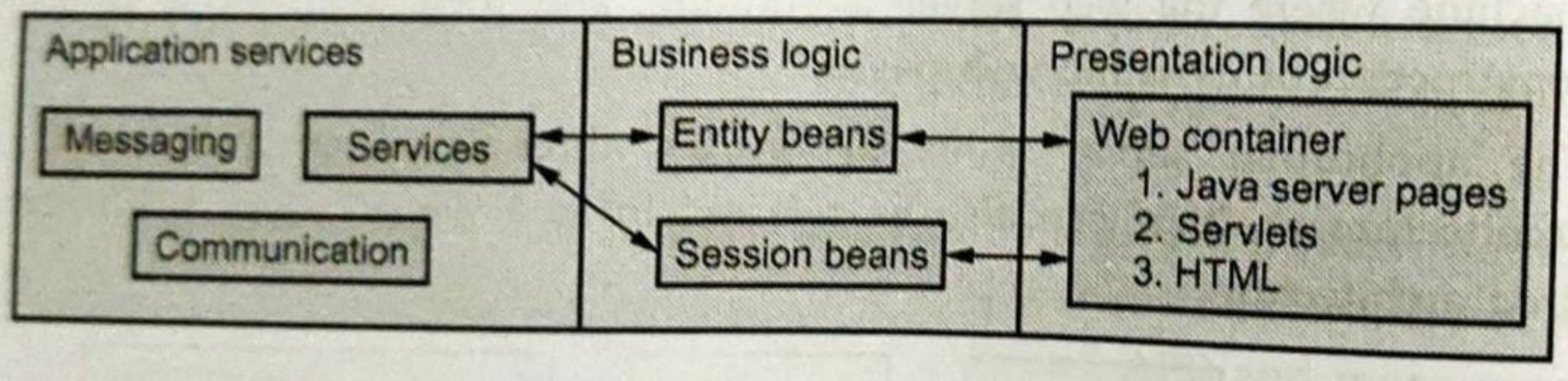


Fig. 1.10.2 : Java application server architecture

1. The Presentation layer requires skills such as HTML, CSS and possibly JavaScript, plus UI design.
2. The Business layer requires skills in a programming language so that business rules can be processed by a computer.
3. The Data Access layer requires SQL skills in the form of Data Definition Language (DDL) and Data Manipulation Language (DML), plus database design.

- Advantages of the 3 Tier Architecture :

1. Flexibility - By separating the business logic of an application from its presentation logic, a 3-Tier architecture makes the application much more flexible to changes.
2. Maintainability - Changes to the components in one layer should have no effect on any others layers.

Scanned by TapScanner

3. Reusability - Separating the application into multiple layers makes it easier to implement re-usable components..
4. Scalability - A 3-Tier architecture allows distribution of application components across multiple servers thus making the system much more scalable.
5. Reliability - A 3-Tier architecture, if deployed on multiple servers, makes it easier to increase reliability of a system by implementing multiple levels of redundancy.

- Fig. 1.10.3 shows two pillars of the web.
- In two pillars of the web, one pillar is HTTP, HTML, URL and application server and web browser is second pillar.
- Communications protocol is a language of digital message formats and rules for exchanging those messages in or between computing systems and/ or in telecommunications. Protocols may include signaling, authentication, and error detection and correction capabilities.

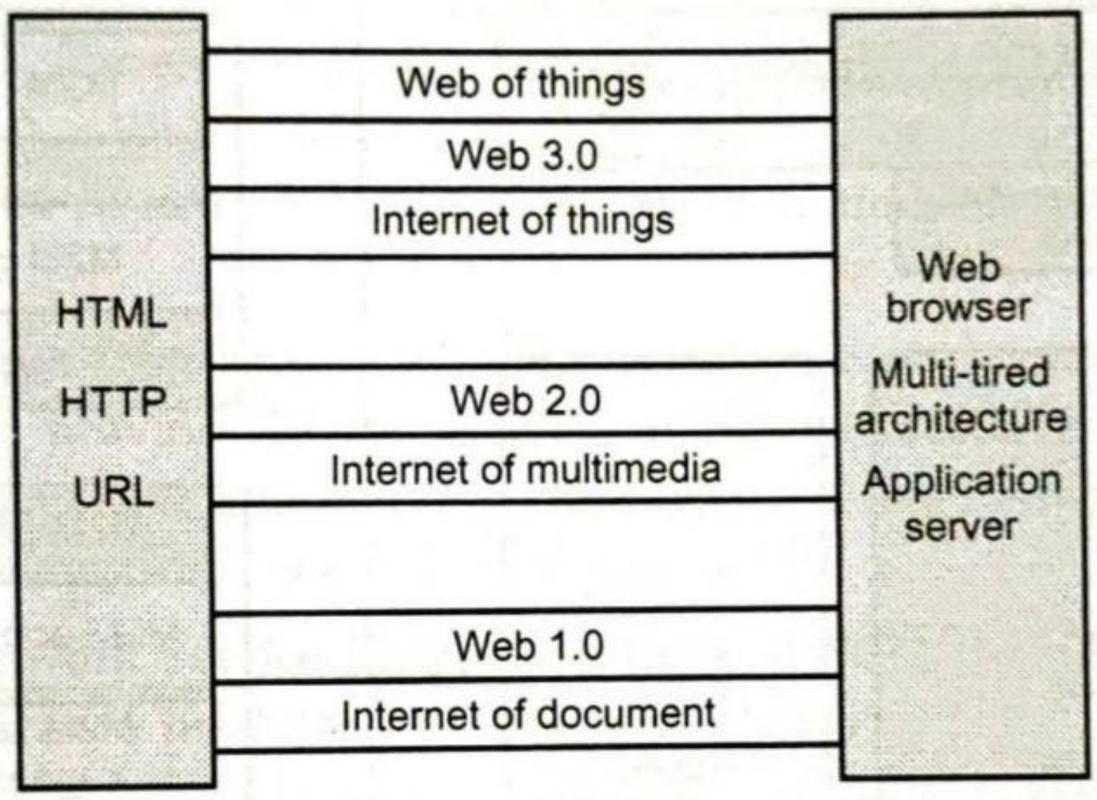


Fig. 1.10.3 : two pillars of the web

1.10.2 Architecture Standardization for WoT

- The W3C Web of Things (WoT) is intended to enable interoperability across IoT Platforms and application domains. Primarily, it provides mechanisms to formally describe IoT interfaces to allow IoT devices and services to communicate with each other, independent of their underlying implementation, and across multiple networking protocols. Secondarily, it provides a standardized way to define and program IoT behavior.
- Day by day, the number of connected Things is growing exponentially. Standardized solutions provided by the rapid evolution of the Internet have laid the foundation of what we call the Web of Things. Therefore, WoT architectures

Scanned by TapScanner

aim to integrate everyday objects with web technologies. Those devices should be able to communicate with each other using existing web standards.

1.10.3 Platform Middleware for IoT

1. Standards for M2M

- The European Telecommunications Standards Institute (ETSI) produces globally-applicable standards for Information and Communications Technologies, including fixed, mobile, radio, converged, broadcast and Internet technologies.
- A new ETSI Technical Committee is developing standards for M2M Communications. This group aims to provide an end-to-end view of M2M standardization cooperating with ETSI's activities on Next Generation Networks and 3GPP standards initiative for mobile communication technologies.
- Fig. 1.10.4 shows the high-level ETSI M2M architecture.

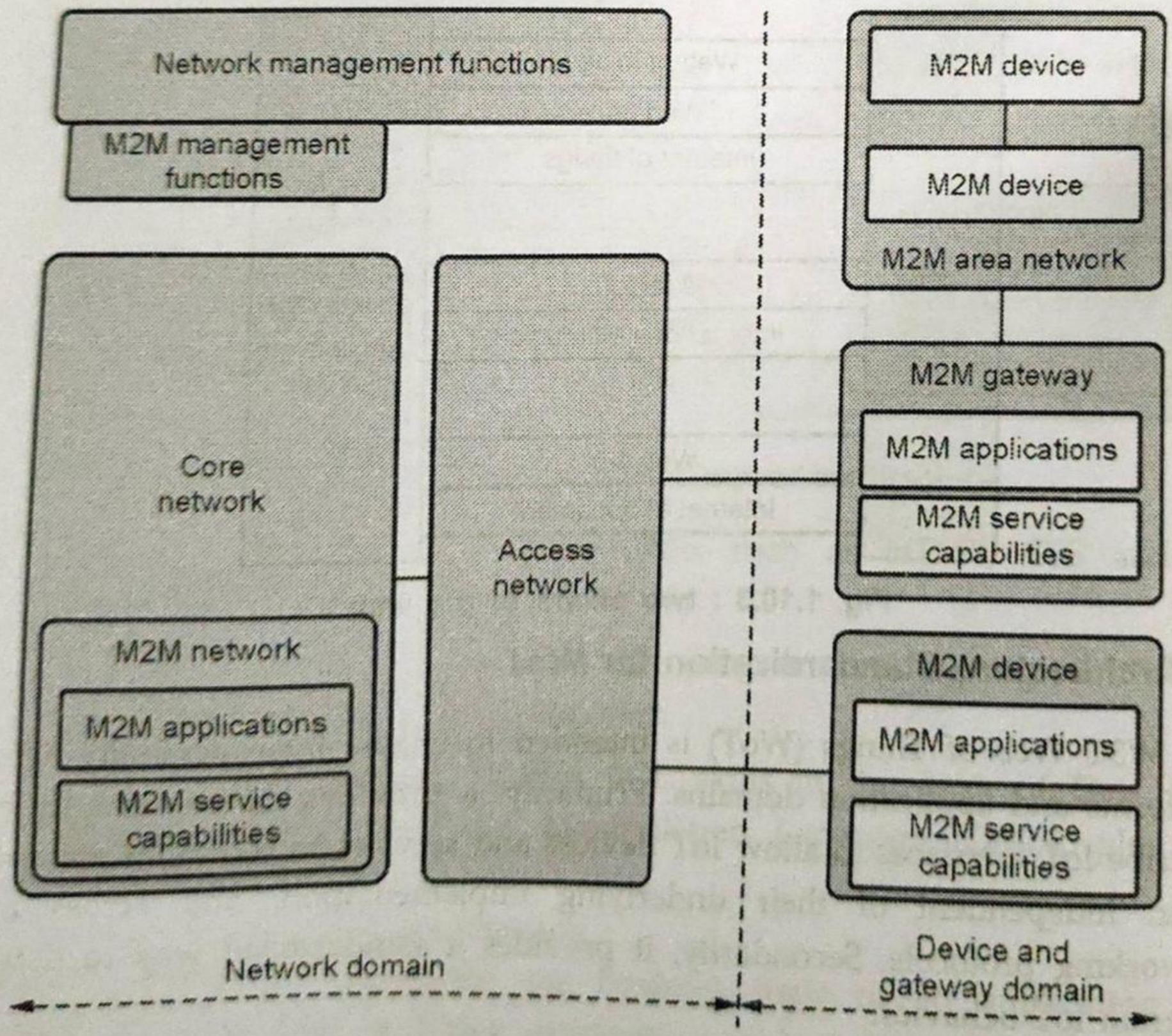


Fig. 1.10.4

Scanned by TapScanner

- A high-level architecture of M2M system consists of a Device and Gateway Domain, and a Network Domain. key elements of the ETSI M2M architecture are as follows :
  1. **M2M Device** runs M2M Device Applications (DA) using M2M Device Service Capabilities Layer (DSCL).
  2. **M2M Gateway** runs M2M Gateway Applications (GA) using M2M Gateway Service Capabilities Layer (GSCL).
  3. **M2M Area Network** provides connectivity based on Personal or Local Area Network technologies (e.g. Zigbee, Bluetooth) between M2M devices and M2M gateways. The case of device-to-device communication is out of the scope of ETSI's effort.
- Network domain contains the following elements :
  1. **M2M Access Network** : It allows M2M devices and M2M gateways to communicate with the core network. It uses any one network solutions: Digital Subscriber Line (DSL), satellite, GSM EDGE Radio Access Network (GERAN), Universal Terrestrial Radio Access Network (UTRAN), evolved UTRAN (eUTRAN), Wi-Fi (IEEE 802.11), and Worldwide Interoperability for Microwave Access (WiMAX), that can be optimized for M2M communication if needed.
  2. **M2M Core Network** : This network enables interconnection with other networks, provides IP connectivity or other connectivity options, service and control functions, and roaming. Similarly to access network, it can be based on varied existing core networking solutions.
  3. M2M Network Service Capabilities Layer (NSCL) : It provides M2M functions that are shared by different M2M applications.
  4. **M2M Applications** run the service logic and use M2M service capabilities available via open interfaces.
  5. **M2M network** management functions consist of all the functions required to manage access and core networks.
  6. M2M management functions consist of all the functions used to facilitate the bootstrapping of permanent M2M service layer security credentials required to manage M2M service capabilities in the network domain.
- The M2M platform middleware normally covers the layers from M2M gateway to the M2M application server. Several (Open Mobile Alliance) OMA standards provide building blocks that map into the ETSI M2M framework :
  1. Device management can provide ETSI's remote entity management service
  2. Gateway management object fulfills some ETSI gateway service requirements
  3. Firmware updates, software updates, provisioning, diagnostics, and monitoring

Scanned by TapScanner

4. Converged personal network services maps into ETSI M2M area network
5. Reachability, address mapping, inter/ intra- area- network messaging, service publication and discovery
6. Some OMA enablers (e.g., location) support services that can be used in M2M applications

2. Frameworks for WSN

- The use of sensor data has become very important in a broad range of applications. Often it is highly desirable to combine sensor data with other kinds of geospatial data in order to build more complex systems. To facilitate the interoperable access to sensor data and thus to allow the flexible (domain independent) re-use of sensor data within spatial data infrastructures a working group of the Open Geospatial Consortium (OGC) has developed the Sensor Web Enablement (SWE) framework.
- SWE is a suite of standard encodings and web services that enable the following :
  1. Discovery of sensors, processors, and observation
  2. Taking of sensors or models
  3. Access to observations and observation streams
  4. Publish-subscribe capabilities for alerts
  5. Robust sensor system and process descriptions
- During the development of the SWE framework, several aims had to be taken into account. Especially the following goals were the drivers of the design of the SWE architecture :
  1. Standardized access to sensor measurements
  2. Retrieval of metadata for determining sensor capabilities and the quality/reliability of measurements
  3. Controlling and tasking of sensors
  4. Alerting based on user defined criteria and sensor measurements
  5. Access to sensor parameters and automatic processing of measurements based upon pre-defined processes
- Web service specifications have been produced by the OGC SWE Working Group are as follows :
  1. Sensor observation service : standard web interface for accessing observations
  2. Sensor planning service : standard web interface for tasking sensor systems and model and requesting acquisitions
  3. Sensor alert service : standard web interface for publishing and subscribing to sensor alerts
  4. Web notification service : standard web interface for asynchronous notification

Scanned by TapScanner

- The USN (Ubiquitous Sensor Networks) standardization of ITU - T is another effort being carried out under the auspices of the Next - Generation Network Global Standards Initiative (NGN - GSI).
- Ubiquitous Sensor Networks is a conceptual framework built over existing physical networks that makes use of sensed data and provide knowledge services. Its main components are as follows :
  1. Sensor Network : Comprising sensors and an independent power source (e.g., battery, solar power). The sensors can then be used for collecting and transmitting information about their surrounding environment;
  2. USN Access Network : Intermediary or "sink nodes" collecting information from a group of sensors and facilitating communication with a control centre or with external entities;
  3. Network Infrastructure : likely to be based on a next-generation network (NGN);
  4. USN Middleware : Software for the collection and processing of large volumes of data;
  5. USN Applications Platform : A technology platform to enable the effective use of a USN in a particular industrial sector or application.

Characteristics of a USN

1. Small-scale sensor nodes;
2. Limited power requirements that can be harvested (e.g., solar power) or stored (e.g., battery);
3. Able to withstand harsh environmental conditions;
4. Fault tolerant and designed to cope with high possibility of node failures;
5. Support for mobility;
6. Dynamic network topology;
7. Able to withstand communication failures;
8. Heterogeneity of nodes;
9. Large scale of deployment

3. Standards for SCADA

- ISO 16100-1:2009, one of the components of ISO 16100 standard for industrial automation systems and controls.

Scanned by TapScanner

- ANSI/ ISA-95 is an international standard for developing an automated interface between enterprise and control systems. This standard is application to global manufacturers.
- A USA ANSI standard developed by an ISA Committee of volunteer experts :
  1. ANSI/ISA 95.01-2000"Enterprise - Control System Integration - Part 1 : Models and Terminology"
  2. ANSI-ISA 95.02-2001"Enterprise - Control System Integration - Part 2 : Object Attributes"
  3. ANSI/ISA 95.03-2005"Enterprise - Control System Integration - Part 3 : Models of Manufacturing Operations"
  4. ISA-95.04, Object Models & Attributes, Part 4: Object models and attributes for Manufacturing Operations Management.
  5. ANSI/ISA 95.05-2007"Enterprise - Control System Integration - Part 5 : Business to Manufacturing Transactions.
- SP95 is the committee developing the ISA95 standard. ISA 95 Level Definitions
  1. Level 0 : Defines the actual physical processes.
  2. Level 1 : Defines the activities involved in sensing and manipulating the physical processes.
  3. Level 2 : Defines the activities of monitoring and controlling the physical processes.
  4. Level 3 : Defines the activities of the work flow to produce the desired end-products.
  5. Level 4 : Defines the business - related activities needed to manage a manufacturing organization.
- Fig. 1.10.5 shows OPC unified architecture.

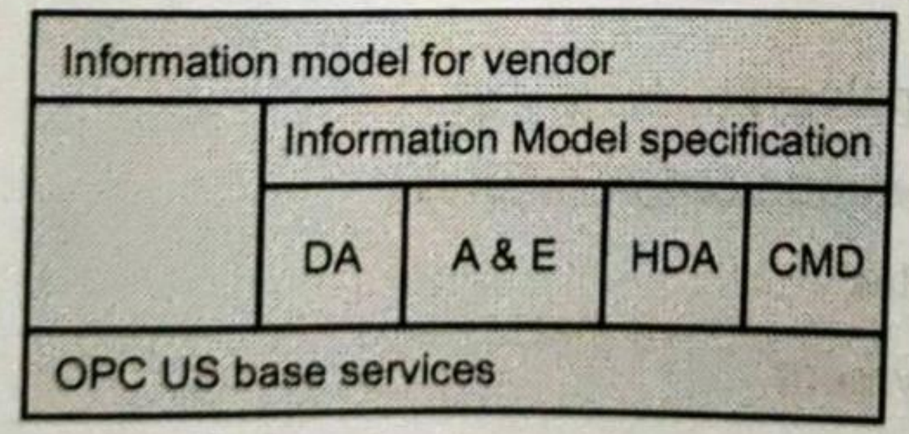


Fig. 1.10.5 : OPC unified architecture

Scanned by TapScanner

1.11 Fill in the Blanks

Q.1 RESTful web service is a web API implemented using _____ and _____ principles.

Q.2 IEEE 802.15.4 is a collection of standards for _____wireless personal area networks.

Q.3 CoAP stands for _____.

Q.4 MQTT is a light weight messaging protocol based on the _____ model.

Q.5 XMPP is a _____ protocol and uses a client-server architecture.

Q.6 _____ APIs allows bi-directional, full duplex communication between clients and servers.

Q.7 A level - 5 IoT system has multiple end nodes and one _____ node.

Q.8 A level - 3 IoT system has a _____ node.

Q.9 AMQP stands for _____.

Q.10 CoAP is a _____ protocol that runs over the UDP for IoT.

Q.11 MQTT is a binary protocol, and it does not support the _____ or _____ of content.

Q.12 The XMPP protocol also uses _____ to bypass firewall barriers.

Q.13 A level _____ IoT system has multiple end nodes and one coordinator node.

Q.14 TCP is the _____ protocol whereas UDP is _____ protocol.

Q.15 IPv6 addresses are _____ bits in length.

Q.16 CoAP uses a _____ architecture where clients communicate with servers using connectionless datagrams.

1.12 Multiple Choice Questions

Q.1 IEEE _____ is a collection of wireless broadband standards, including extensive descriptions for the link layer.

a 802.3 b 802.11

c 802.15.4 d 802.16

Q.2 Representational State Transfer APIs follow the _____ communication model.

a request-response b publish-subscribe

c push-pull d exclusive pair

Scanned by TapScanner

Q.3 XMPP stands for _____.

a EXtensible Markup Presence Protocol

b Extensible Messaging Presence Protocol

c Extensible Messaging Push-Pull

d Extensible Messaging Protocol Publish

Q.4 _____ is a lightweight broker-based publish / subscribe messaging protocol designed to be open, simple, lightweight and easy to implement.

a XMPP

b AMQP

c MQTT

d DDS

Q.5 _____ model is stateless communication model and each request-response pair is independent of others.

a Exclusive pair

b Request-response

c Publish-subscribe

d Push-pull

Q.6 _____ is an open application layer protocol for business messaging.

a MQTT

b DDS

c AMQP

d Websocket

Q.7 _____ is a web protocol that runs over the UDP for IoT.

a UPnP

b CoAP

c MQTT

d HTTP

Q.8 Which protocol is lightweight ?

a MQTT

b HTTP

c CoAP

d None of the these

Q.9 _____ is a peer-to-peer network architecture for connecting intelligent appliances, wireless devices, and PCs that are in close proximity.

a XMPP

b UART

c UPnP

d SPI

Scanned by TapScanner

Q.10 Which of the following is not IoT servies as a platform ?

a SenseIoT
b Clayster
c Thinger.io
d XMPP

Q.11 _____ is a web protocol that runs over the UDP for IoT.

a UPnP
b CoAP
c MQTT
d HTTP

Q.12 MQTT stands for _____.

a Message Queue Transport Telemetry
b Machine Queue Telemetry Transport
c Message Queue Telemetry Transport
d Message Queue Transfer Transport

Q.13 List the 3I characteristics of the Internet of Things.

a Instrumented
b Interconnected
c Intelligently
d All of these

Q.14 Which of the following is NOT an IoT enabling technology ?

a Cloud computing
b Big data analytic
c Embedded system
d LAN

Q.15 CoAP uses _____ protocol.

a TCP
b UDP
c TCP and UDP
d IP

Q.16 HTTP protocol is used by _____ model.

a request/response
b publish/subscribe
c push/pull
d none

Q.17 Google AppEngine is an example of a _____.

a IaaS
b PaaS
c IaaS and PaaS
d None

Scanned by TapScanner

Q.18 IoT stands for _____.

a Intranet of things
b Internet of Thoughts
c Internet of Things
d Information of Things

Q.19 The _____ in IoT usually refers to IoT devices which have unique identities and can perform remote sensing, actuating and monitoring capabilities.

a things
b information
c protocol
d data

Q.20 _____ model is stateless communication model and each request-response pair is independent of others.

a Exclusive pair
b Request-response
c Publish-subscribe
d Push-pull

Q.21 IoT communication models are _____.

a request/response model
b publish/subscribe model
c push/pull model
d all of the above

Q.22 A level _____ IoT system has multiple independent end nodes that perform sensing and/or actuation and send data to the cloud.

a 2
b 3
c 5
d 6

Q.23 A level-3 IoT system has a _____ node.

a multiple
b single
c end
d independent

Scanned by TapScanner

Answer Keys for Fill in the Blanks

Q.1	HTTP, REST	Q.2	low rate	Q.3	Constrained Application Protocol
Q.4	publish subscribe	Q.5	decentralized	Q.6	webSocket
Q.7	coordinator	Q.8	single	Q.9	Advanced Message Queuing Protocol
Q.10	web	Q.11	encoding, decoding	Q.12	message brokers
Q.13	5	Q.14	connection oriented, connectionless	Q.15	128
Q.16	client - server				

Answer Keys for Multiple Choice Questions

Q.1	d	Q.2	a	Q.3	b	Q.4	c
Q.5	b	Q.6	c	Q.7	b	Q.8	a
Q.9	c	Q.10	d	Q.11	b	Q.12	c
Q.13	d	Q.14	d	Q.15	b	Q.16	a
Q.17	b	Q.18	c	Q.19	a	Q.20	b
Q.21	d	Q.22	d	Q.23	b		

❑❑❑

Scanned by TapScanner