

Unit -1

Internet of Things

IoT in our daily lives...





Introduction

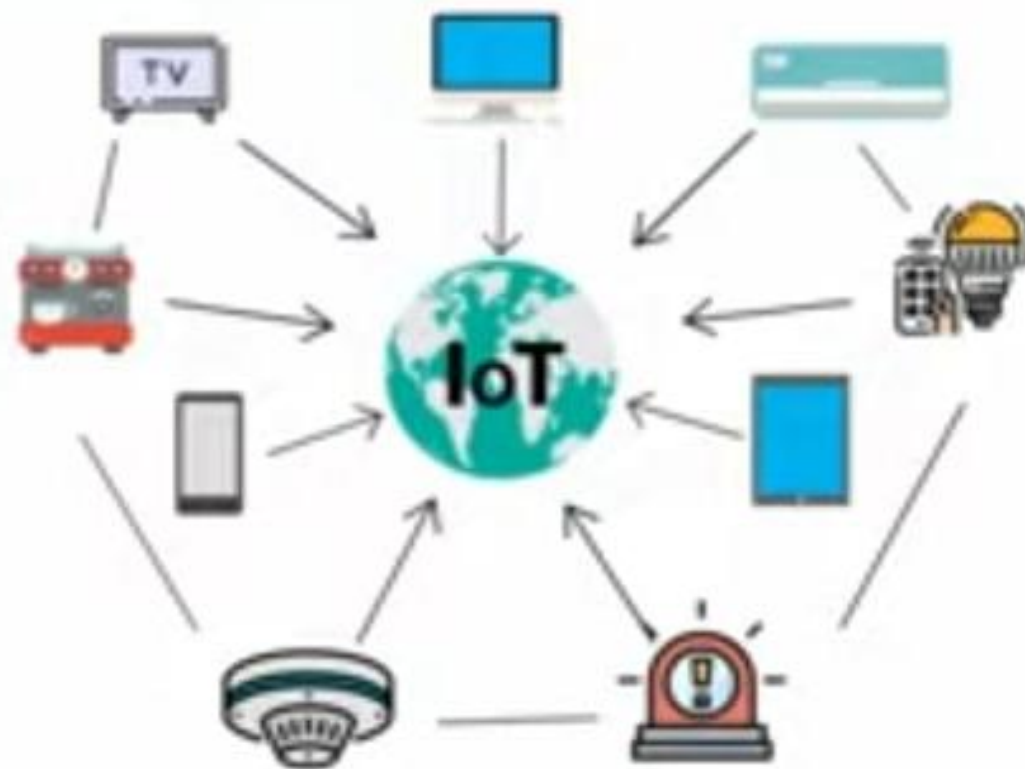
- Imagine a person named joy who has a smart home equipped with IoT devices. Among these devices is a smart air conditioning (A/C) system connected to the internet.
- One day, joy leaves for work and forgets to turn off the A/C at home. Instead of worrying about the energy consumption and comfort levels, joy can use a mobile application on his smartphone to remotely control the A/C.
- Because The A/C unit at joy 's home is a smart device equipped with IoT capabilities. It is connected to the internet, allowing it to receive and process commands remotely.

A smart home is the best example of IoT



In the Internet of Things (IoT) ecosystem, interrelated devices communicate with each other to collect, exchange, and act upon data. This communication is essential for creating a smart and interconnected environment.





IoT is changing the way we live our lives



❑ Definition

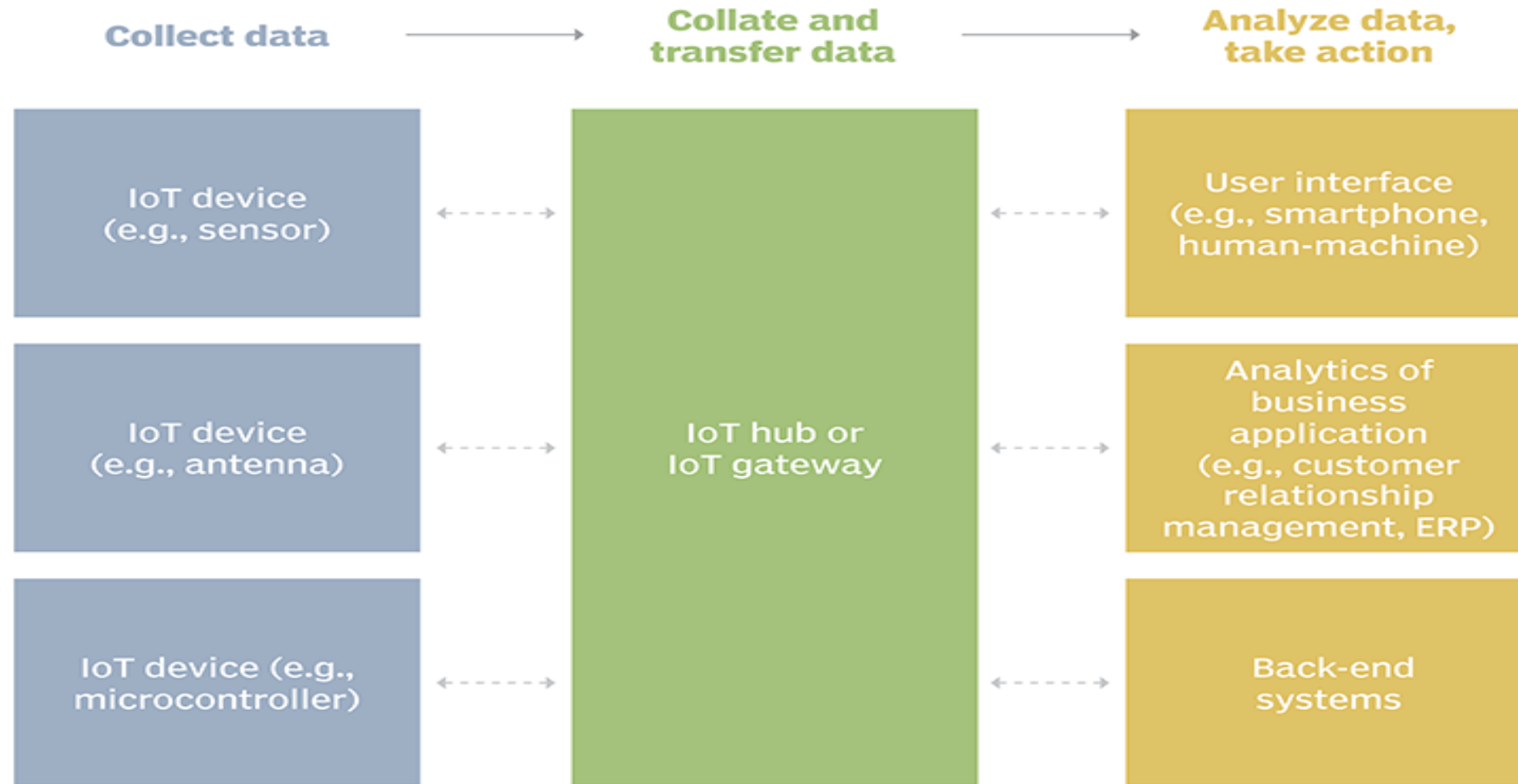
- **Internet of Things (IoT) is a concept which enables communication between internetworking devices and applications, whereby physical objects or ‘things’ communication through the Internet.**



❑ Definition

- IoT means a network of
 - physical things (objects) sending, receiving, or communicating information using the Internet or other communication technologies and network just as the computers, tablets and mobile and thus, enabling the monitoring, coordinating or controlling process across the Internet or another data network
- IoT is the network of physical objects or ‘things’
 - That embedded with electronics, software, sensors and connectivity to enable it to achieve greater value and service by exchanging data with the manufacturer, operator and/or other connected devices.

Example of an IoT system



□ Vision



- The concept of IoT enables, for example, GPS- based tracking, controlling and monitoring of devices, machine- to- machine(M2M) communication, connected cars, communication between wearable and personal devices and Industry.
- The vision of IoT can be understood through Examples 1.1 and 1.2
- Example 1.1 : Umbrella
- Example 1.2 : Streetlights

IoT Example 1.1



Example 1.1

Through computing, an umbrella can be made to function like a living entity. By installing a tiny embedded device, which interacts with a web based weather service and the devices owner through the Internet the following communication can take place. The umbrella, embedded with a circuit for the purpose of computing and communication connects to the Internet. A website regularly publishes the weather report. The umbrella receives these reports each morning, analyses the data and issues reminders to the owner at intermittent intervals around his/her office-going time. The reminders can be distinguished using differently coloured LED flashes such as red LED flashes for hot and sunny days, yellow flashes for rainy days.

A reminder can be sent to the owner's mobile at a pre-set time before leaving for office using NFC, Bluetooth or SMS technologies. The message can be—(i) *Protect yourself from rain. It is going to rain. Don't forget to carry the umbrella;* (ii) *Protect yourself from the sun. It is going to be hot and sunny. Don't forget to carry the umbrella.* The owner can decide to carry or not to carry the umbrella using the Internet connected umbrella.

IoT Example 1.2



Example 1.2

Streetlights in a city can be made to function like living entities through sensing and computing using tiny embedded devices that communicate and interact with a central control-and-command station through the Internet. Assume that each light in a group of 32 streetlights comprises a sensing, computing and communication circuit. Each group connects to a group-controller (or coordinator) through Bluetooth or ZigBee. Each controller further connects to the central command-and-control station through the Internet.

The station receives information about each streetlight in each group in the city at periodic intervals. The information received is related to the functioning of the 32 lights, the faulty lights, about the presence or absence of traffic in group vicinity, and about the ambient conditions, whether cloudy, dark or normal daylight.

The station remotely programs the group controllers, which automatically take an appropriate action as per the conditions of traffic and light levels. It also directs remedial actions in case a fault develops in a light at a specific location. Thus, each group in the city is controlled by the 'Internet of streetlights'. Figure 1.1 shows the use of the IoT concept for streetlights in a city.

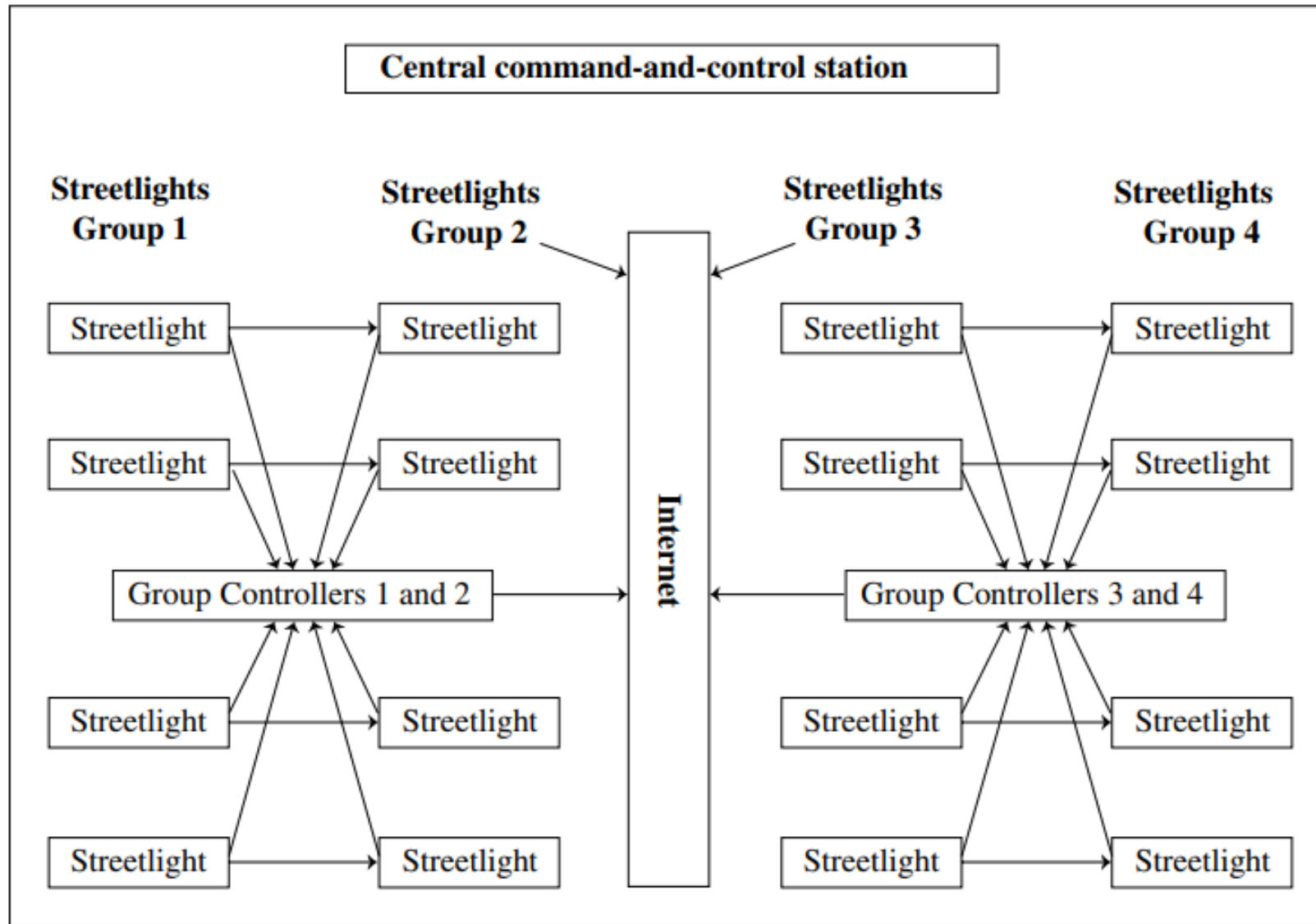


Figure 1.1 Use of Internet of Things concept for streetlights in a city

Characteristics Of IoT



1. Connectivity

- Connectivity is an important requirement of the IoT infrastructure.
- Things of IoT should be connected to the IoT infrastructure.
- Anyone, anywhere, anytime can connect, this should be guaranteed at all times.
- For example, the connection between people through Internet devices like mobile phones, and other gadgets, also a connection between Internet devices such as routers, gateways, sensors, etc.



2. Intelligence and Identity

- The extraction of knowledge from the generated data is very important.
- For example, a sensor generates data, but that data will only be useful if it is interpreted properly. Each IoT device has a unique identity. This identification is helpful in tracking the equipment and at times for querying its status.



3. Scalability

- The number of elements connected to the IoT zone is increasing day by day. Hence, an IoT setup should be capable of handling the massive expansion. The data generated as an outcome is enormous, and it should be handled appropriately.

4. Dynamic and Self-Adapting (Complexity)

- IoT devices should dynamically adapt themselves to changing contexts and scenarios. Assume a camera meant for surveillance. It should be adaptable to work in different conditions and different light situations (morning, afternoon, and night).



5. Safety

Sensitive personal details of users are at risk of being compromised when all their devices are connected to the internet, potentially leading to significant losses. Therefore, ensuring data security is a major challenge. Additionally, the extensive equipment involved poses further risks, and IoT networks may also be vulnerable. As a result, ensuring the safety of equipment is equally critical.



IoT CONCEPTUAL FRAMEWORK

The Internet of Things (IoT) conceptual framework is a structured ecosystem that explains how data moves from sensors to analysis and how the physical and digital worlds are connected. It includes:

- **Data flow:** How data moves from IoT devices and services to the cloud server
- **Data gathering:** Gather and consolidate raw data from a sensor network and gateways
- **Data communication:** Communicate with each other via data streams
- **Data management:** Subsystems for connection, assembly, collection, and management
- **Data analysis:** Subsystems for real-time data analysis, series analysis, and intelligence
- **Data storage:** A data store, database, or cloud infrastructure



IoT CONCEPTUAL FRAMEWORK

- Example 1.1 showed a single object (umbrella) communicating with a central server for acquiring data.
- The following equation describes a simple conceptual framework of IoT:

**Physical Object + Controller, Sensor and Actuators +
Internet = Internet of Things**1.1



Physical Object + Controller, Sensor and Actuators + Internet = Internet of Things1.1

- Equation 1.1 conceptually describes the Internet of umbrellas, a controller, sensor and actuators, and the Internet for connectivity to a web service and a mobile service provider.
- Generally, IoT consists of an internetwork of devices and physical objects wherein a number of objects can gather the data at remote locations and communicate to units managing, acquiring, organising and analyzing the data in the processes and services.



- Example 1.2 showed the number of streetlights communicating data to the group controller which connects to the central server using the Internet. A general framework consists of the number of devices communicating data to a data centre or an enterprise or a cloud server.
- The IoT framework, utilized in numerous applications as well as in enterprise and business processes, is generally more complex than the one represented by Equation 1.1. The following equation conceptually illustrates the actions and data communication occurring at successive levels in IoT, involving interconnected devices and objects.



Gather + Enrich + Stream + Manage + Acquire + Organise
and Analyse = Internet of things with connectivity to
data centre, enterprise or cloud server

....1.2

Equation 1.2 is an IoT conceptual framework for the
enterprise processes and services

Gather + Enrich + Stream + Manage + Acquire + Organize and Analyze = Internet of things with connectivity to data center, enterprise or cloud server1.2

Gather: IoT devices collect data using sensors (e.g., temperature, humidity, motion).

Enrich: The raw data is enhanced by adding context or combining it with other information to make it more useful.

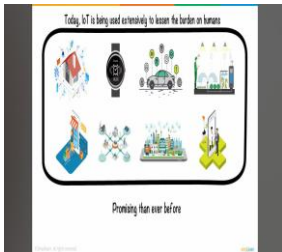
Stream: The data is transmitted, often in real time, to a central system or edge devices for processing.

Manage: The system ensures proper control of devices, updates software, and monitors performance.

Acquire: Data is securely stored in a system (e.g., cloud, local servers) for future use.

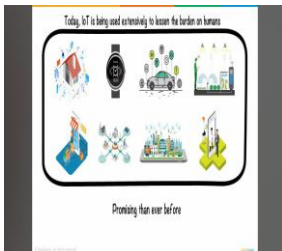
Organize and Analyze: The stored data is processed and analyzed to generate insights and inform decisions.

Connectivity: IoT systems connect to data centers, enterprise solutions, or cloud servers for advanced processing, remote access, and scalability.



Gather + Enrich + Stream + Manage + Acquire + Organize and Analyze =
Internet of things with connectivity to data center, enterprise or cloud
server1.2

The diagram represents how IoT systems operate as a closed loop of collecting, transmitting, managing, and analyzing data to drive meaningful actions and insights. This interconnected flow ensures seamless communication between devices and centralized systems like the cloud or enterprise platforms, enabling efficient decision-making and automation in various domains like healthcare, manufacturing, transportation, and smart homes.



Gather + Consolidate + Connect + Collect + Assemble + Manage and Analyse= Internet of Things with connectivity and cloud server ...1.3

- The equation above is an alternative conceptual representation for a complex system.
- It is based on IBM IoT conceptual framework. The equation shows the actions and communication of data at successive levels in IoT. The framework manages the IoT services using data from internetwork of the devices and objects, internet and cloud services, and represents the flow of data from the IoT devices for managing the IoT services using the cloud server.



Gather + Consolidate + Connect + Collect + Assemble + Manage and Analyse= Internet of Things with connectivity and cloud server ...1.3

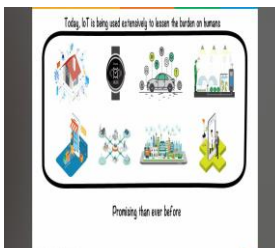


Equation 1.3 represents a complex conceptual framework for IoT using cloud-platform based processes and services. The steps are as follows:

1. Levels 1 and 2 consist of a sensor network to gather and consolidate the data. First level gathers the data of the things (devices) using sensors circuits. The sensor connects to a gateway. Data then consolidates at the second level, for example, transformation at the gateway at level 2.
2. The gateway at level 2 communicates the data streams between levels 2 and 3. The system uses a communication-management subsystem at level 3

Gather + Consolidate + Connect + Collect + Assemble
+ Manage and Analyse= Internet of Things with
connectivity and cloud server ...1.3

3. An information service consists of connect, collect, assemble and manage subsystems at levels 3 and 4. The services render from level 4.
4. Real time series analysis, data analytics and intelligence subsystems are also at levels 4 and 5. A cloud infrastructure, a data store or database acquires the data at level 5.



1. Levels 1 and 2 consist of a **sensor network to gather and consolidate the data**. First level gathers the data of the things (devices) using sensors circuits. The sensor connects to a gateway. Data then consolidates at the second level, for example, **transformation at the gateway** at level 2.
2. The gateway at level 2 communicates the data streams between levels 2 and 3.
3. The system uses a **communication-management subsystem** at level 3.

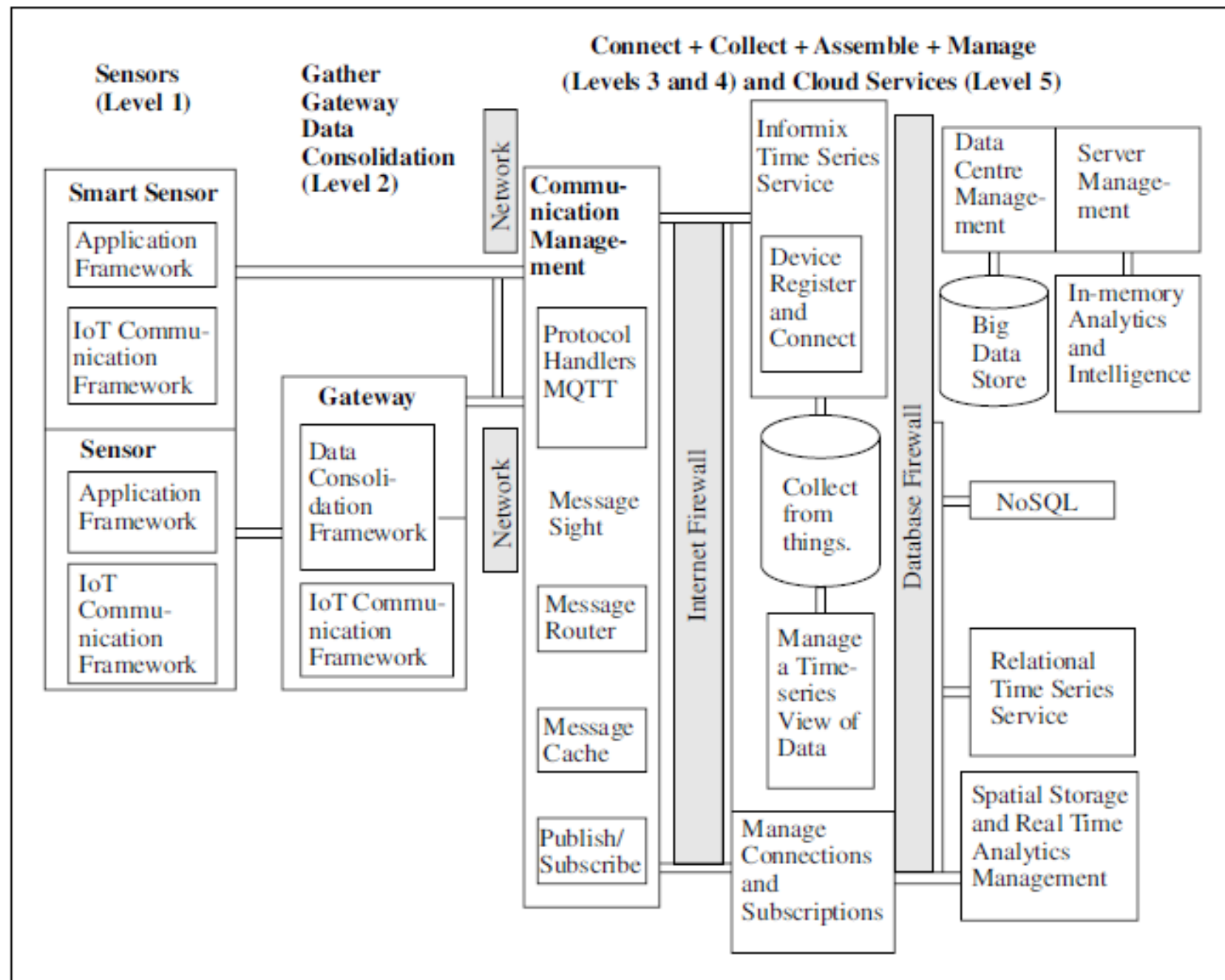


Figure 1.3 IBM IoT conceptual framework

3. An information service consists of **connect, collect, assemble and manage subsystems** at levels 3 and 4. The services render from level 4.

4. Real time **series analysis, data analytics and intelligence subsystems** are also at levels 4 and 5. A cloud infrastructure, a data store or database **acquires the data** at level 5.

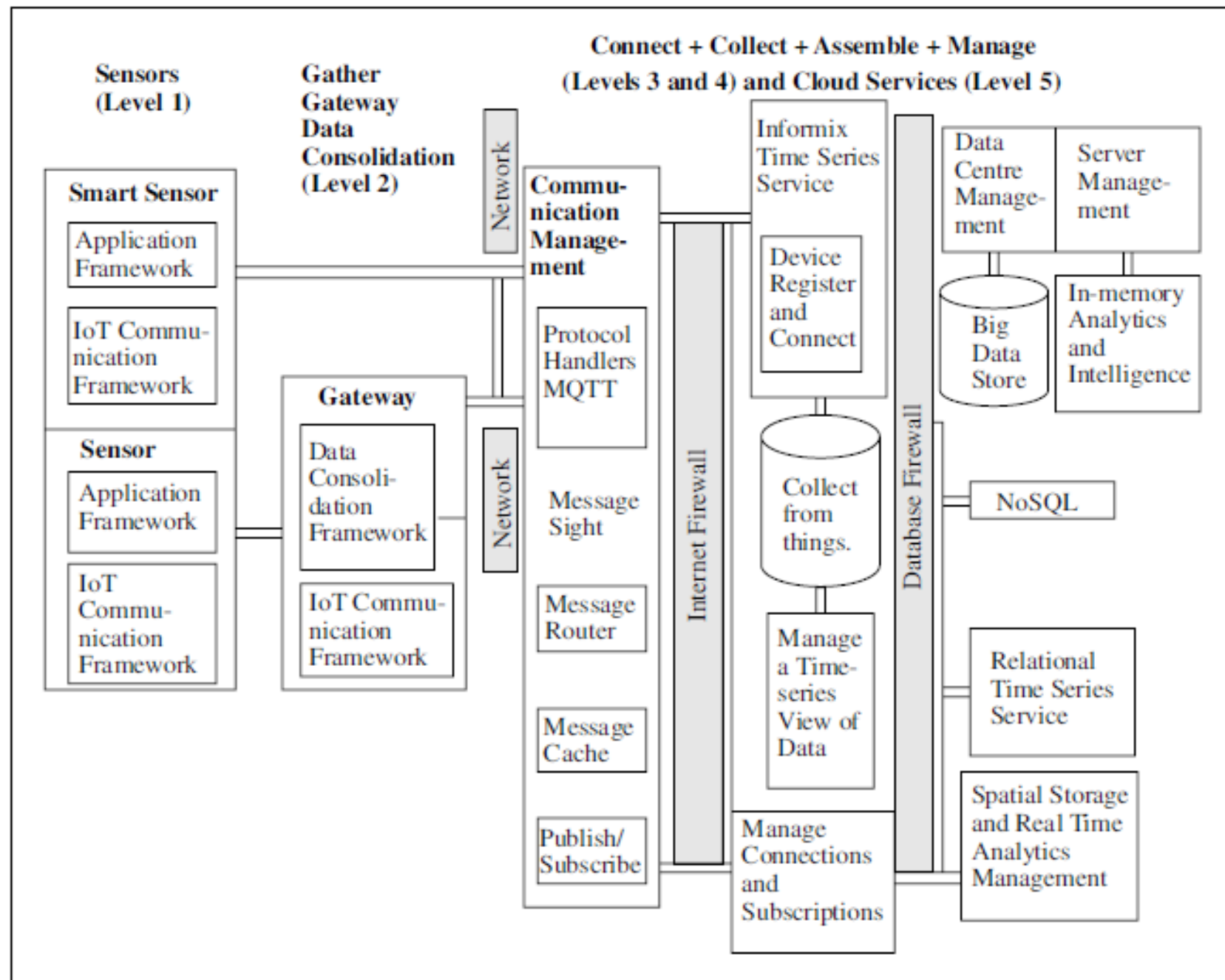


Figure 1.3 IBM IoT conceptual framework

Reconfirm Your Understanding

- *Adrian McEwen and Hakim Cassimally* equation is a simple conceptualisation of a framework for IoT with connectivity to a web service:

Physical Object + Controller, Sensor and Actuators + Internet = Internet of Things

- An equation to conceptualise a general framework for IoT with connectivity to a data centre, application or enterprise server for data storage, services and business processes is:

**Gather + Enrich + Stream + Manage + Acquire + Organise and Analyse
= Internet of Things**

Oracle suggested IoT architecture is the basis for this equation.

- Another equation which conceptualises the general framework for IoT using the cloud based services is:

**Gather + Consolidate + Connect + Collect + Assemble + Manage and Analyse
= Internet of Things**



ARCHITECTURAL VIEW OF IOT

- Architecture consists of different layers of technologies supporting IoT.
- It serves to illustrate how various technologies relate to each other and to communicate the scalability, modularity and configuration of IoT deployments in different scenarios.
- An IoT system has multiple levels (Equations 1.1 to 1.3). These levels are also known as tiers. A model enables conceptualisation of a framework. A reference model can be used to depict building blocks, successive interactions and integration.

An example is CISCO's presentation of a reference model comprising seven levels (Figure 1.4)

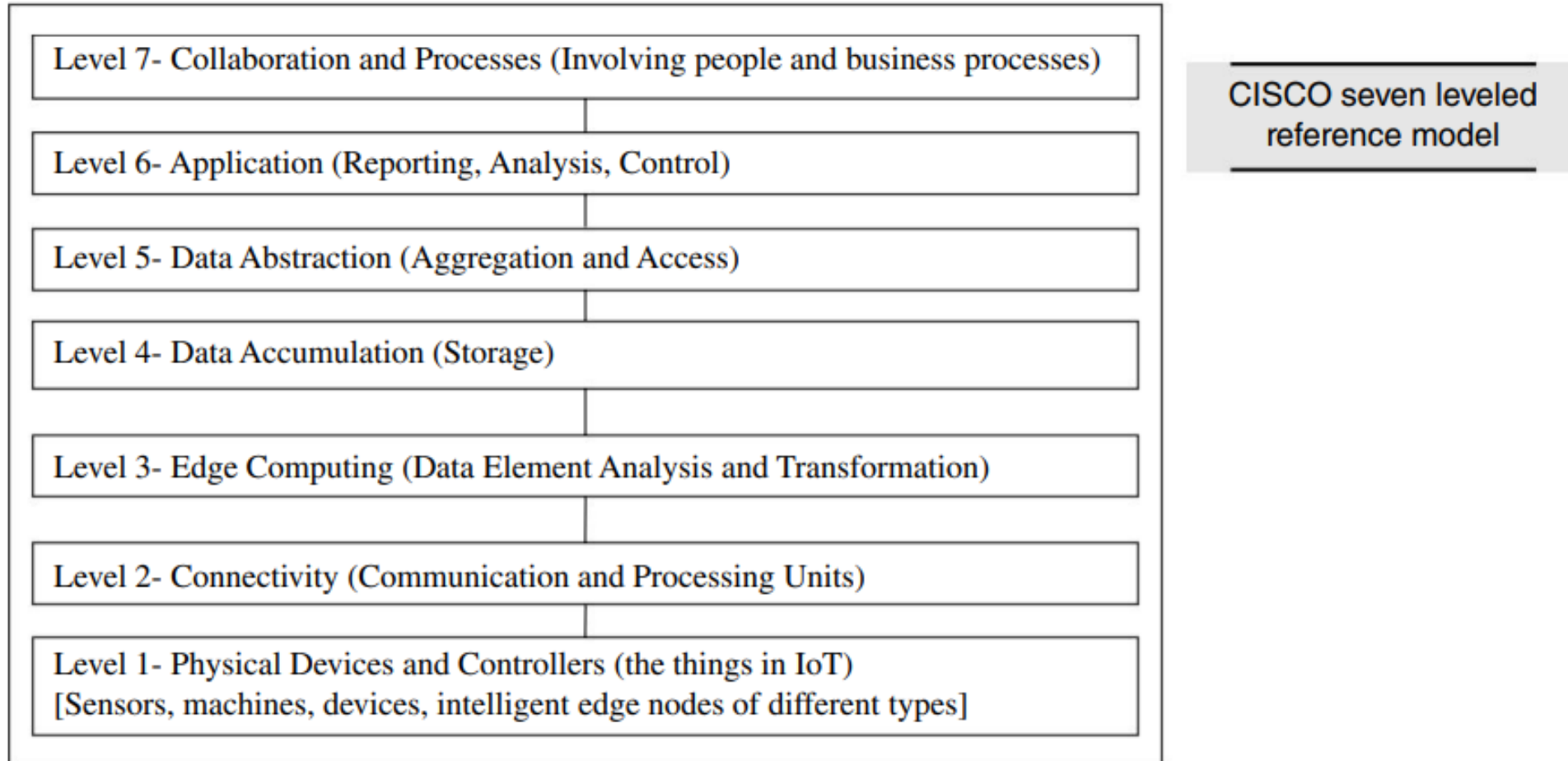


Figure 1.4 An IoT reference model suggested by CISCO that gives a conceptual framework for a general IoT system

The functionality of each layer is described below:

➤ Smart device / sensor layer:



- The lowest layer is made up of smart objects integrated with sensors.
- The sensors enable the interconnection of the physical and digital worlds allowing real-time information to be collected and processed. There are various types of sensors for different purposes.
- The sensors have the capacity to take measurements such as temperature, air quality, speed, humidity, pressure, flow, movement and electricity etc. In some cases, they may also have a degree of memory, enabling them to record a certain number of measurements.
- A sensor can measure the physical property and convert it into signal that can be understood by an instrument. Sensors are grouped according to their unique purpose such as environmental sensors, body sensors, home appliance sensors and vehicle telemetric sensors, etc.

➤ Gateways and Networks

- Massive volume of data will be produced by these tiny sensors and this requires a robust and high performance wired or wireless network infrastructure as a transport medium.
- Current networks, often tied with very different protocols, have been used to support machine-to-machine (M2M) networks and their applications.
- With demand needed to serve a wider range of IoT services and applications such as high speed transactional services, context- aware applications, etc, multiple networks with various technologies and access protocols are needed to work with each other in a heterogeneous configuration.
- These networks can be in the form of a private, public or hybrid models and are built to support the communication requirements for latency, bandwidth or security. Various gateways (microcontroller, microprocessor) & gateway networks (WI-FI (Wireless Fidelity), GSM (Global System for Mobile Communication), GPRS (General Packet Radio Service)).

➤ **Management Service Layer/ Data Processing Layer**

- **Data Aggregation and Processing:** Collects, filters, and processes data from IoT devices to derive actionable insights.
- **Analytics and Decision Support:** Applies analytics and AI techniques to enable real-time and batch decision-making.
- **Service Management:** Manages IoT devices, including monitoring, updates, and provisioning, ensuring smooth operations.
- **Security and Privacy:** Ensures secure data handling with encryption, access control, and compliance with privacy regulations.

Management Service Layer/ Data Processing Layer

- The data processing layer of IoT architecture refers to the software and hardware components that are responsible for collecting, analyzing, and interpreting data from IoT devices.
- This layer is responsible for receiving raw data from the devices, processing it, and making it available for further analysis or action.
- The data processing layer includes a variety of technologies and tools, such as data management systems, analytics platforms, and machine learning algorithms.
- These tools are used to extract meaningful insights from the data and make decisions based on that data.

➤ **Application Layer**

- The application layer of IoT architecture is the topmost layer that interacts directly with the end-user.
- It is responsible for providing user-friendly interfaces and functionalities that enable users to access and control IoT devices.
- This layer includes various software and applications such as mobile apps, web portals, and other user interfaces that are designed to interact with the underlying IoT infrastructure.
- The IoT application covers “smart” environments/spaces in domains such as: Transportation, Building, City, Lifestyle, Retail, Agriculture, Factory, Supply chain, Emergency, Healthcare, User interaction, Culture and tourism, Environment and Energy.

TECHNOLOGY BEHIND IoT

- **Hardware** (Arduino Raspberry Pi, Intel Galileo, Intel Edison, ARM mBed, Bosch XDK110, Beagle Bone Black and Wireless SoC)
- **Integrated Development Environment** (IDE) for developing device software, firmware and APIs
- **Protocols** [RPL, CoAP, RESTful HTTP, MQTT, XMPP (Extensible Messaging and Presence Protocol)]
- **Communication** (Powerline Ethernet, RFID, NFC, 6LowPAN, UWB, ZigBee, Bluetooth, WiFi, WiMax, 2G/3G/4G)
- **Network backbone** (IPv4, IPv6, UDP and 6LowPAN)
- **Software** (RIOT OS, Contiki OS, Thingsquare Mist firmware, Eclipse IoT)
- **Internet Cloud Platforms/Data Centre** (Sense, ThingWorx, Nimbits, Xively, openHAB, AWS IoT, IBM BlueMix, CISCO IoT, IOx and Fog, EvryThng, Azure, TCS CUP)

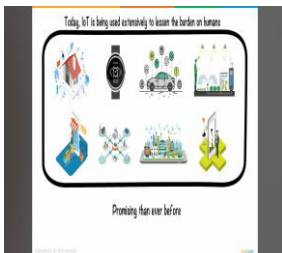
<https://youtu.be/5NfKeofbac4>

1.4.1 Server-end Technology

IoT servers are application servers, enterprise servers, cloud servers, data centres and databases.

Servers offer the following software components:

- Online platforms
- Devices identification, identity management and their access management
- Data accruing (process of collecting, transmitting, and managing data generated by IoT devices and sensors within a network), aggregation, integration, organising and analysing
- Use of web applications, services and business processes



1.4.2 Major Components of IoT System



Physical object

- with embedded software into a hardware

Hardware

- consisting of a microcontroller, firmware, sensors, control unit, actuators and communication module.

Communication module

- Software consisting of device APIs and device interface for communication over the network and communication circuit/port(s), and middleware for creating communication stacks using 6LowPAN, CoAP, LWM2M, IPv4, IPv6 and other protocols.

Software

- for actions on messages, information and commands which the devices receive and then output to the actuators, which enable actions such as glowing LEDs, robotic hand movement etc.

Sensors and Control Units



Sensors:

- Sensors are electronic devices that sense the physical environments.
- An industrial automation system or robotic system has multiple smart sensors embedded in it. Sensor-actuator pairs are used in control systems. A smart sensor includes computing and communication circuits.

Sensors are of two types.

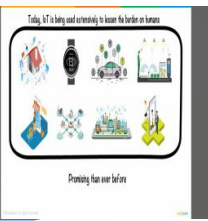
1. The first type gives analog inputs to the control unit.

- Analog input is a **continuous input from the field**
- Examples are thermistor, photoconductor, pressure gauge and Hall sensor.

2. The second type gives digital inputs to the control unit.

- Examples are touch sensor, proximity sensor, metal sensor, traffic presence sensor, rotator encoder for measuring angles and linear encoders for measuring linear displacements.

Control Units



- Most commonly used control unit in IoT consists of a Microcontroller Unit (MCU) or a custom chip.
- A microcontroller is an integrated chip or core in a VLSI (Very Large Scale Integration) or SoC (Security Operations Center).
- Popular microcontrollers are ATmega 328, ATmega 32u4, ARM Cortex and ARM LPC (Advanced RISC Machine, Low Pin Count) .

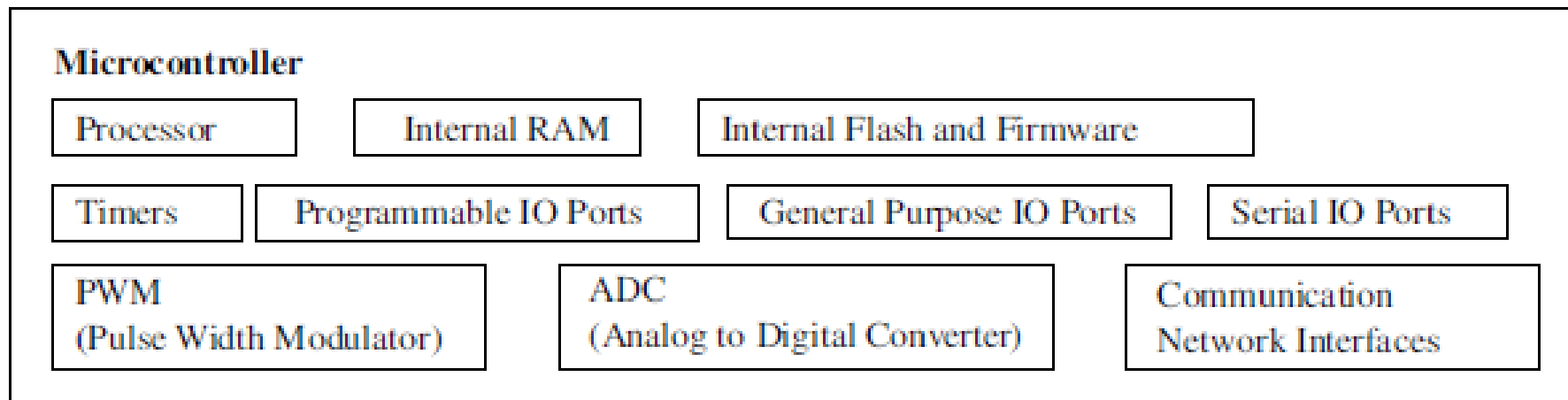


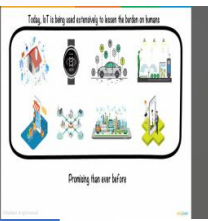
Figure 1.6 Various functional units in an MCU that are embedded in an IoT device or a physical object



Communication Module

- A communication module consists of protocol handlers, message queue and message cache.
- This is a data structure or mechanism used to temporarily store and manage messages (data units) for orderly processing.
- Temporarily stores frequently accessed or recently processed messages to speed up access and reduce redundant processing..

Software



- IoT software consists of two components—software at the IoT device and software at the IoT server.
- Figure 1.7 shows the software components for the IoT device hardware and server.

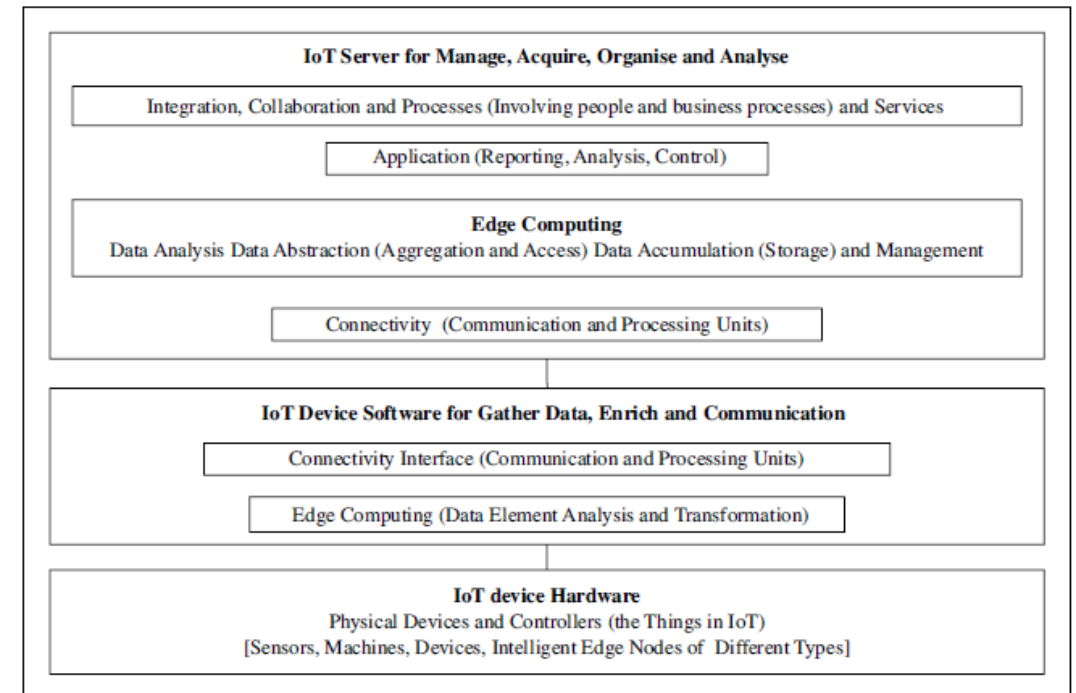


Figure 1.7 IoT software components for device hardware

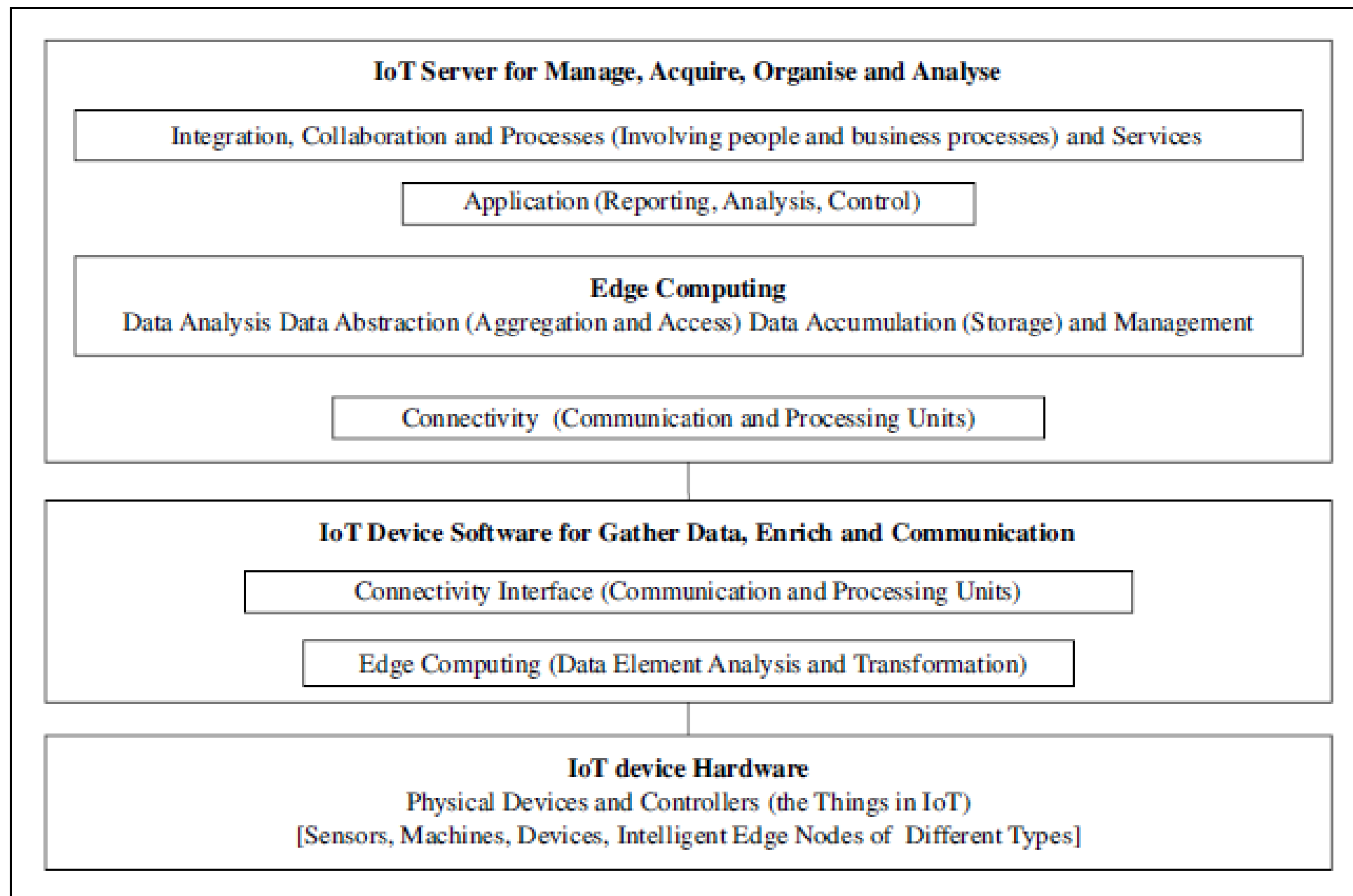
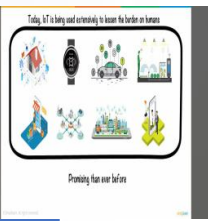


Figure 1.7 IoT software components for device hardware

Middleware



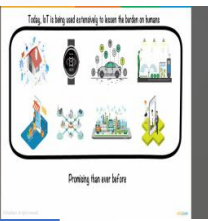
- OpenIoT is an open-source middleware platform designed to enable the integration, management, and utilization of Internet of Things (IoT) devices and services.
- It provides a framework for connecting IoT devices, collecting sensor data, and delivering it to applications or users through cloud-based solutions.
- IoT SyS is middleware that helps smart devices communicate by supporting various standards and protocols, including IPv6, oBIX, 6LoWPAN, and CoAP. It simplifies connecting and managing smart devices by providing a unified communication system.
- The oBIX is standard XML and web services protocol oBIX (Open Building Information Xchange).



Operating Systems (OS)

- Examples of OSs are RIOT, Raspbian, AllJoyn, Spark and Contiki.
- RIOT is an operating system for IoT devices.
- Raspbian is a popular Raspberry Pi operating system
- AllJoyn is an open-source OS created by Qualcomm.
- Spark is a distributed, cloud-based IoT operating system and web-based IDE.
- Contiki OS7 is an open-source multitasking OS.

Firmware



- Thingsquare Mist is an open-source firmware (software embedded in hardware) for true Internet-connectivity to the IoT.
- It enables resilient wireless mesh networking.
- Several microcontrollers with a range of wireless radios support Things MIST.
- Firmware is a form of microcode or program embedded into hardware devices to help them operate effectively. Hardware like cameras, mobile phones, network cards, optical drives, printers, routers, scanners, and television remotes rely on firmware built into their memory to function smoothly.

Let us categorise the resources
which enable the development
of IoT prototype and product

SOURCES OF IoT

- Examples of hardware sources for IoT prototype development are:
 1. Arduino Yún (ATmega32U4 microcontroller)
 2. Microduino
 3. Beagle Board
 4. RasWIK, etc.
- Hardware prototype needs an IDE(integrated Development Environment) for developing device software, firmware and APIs.

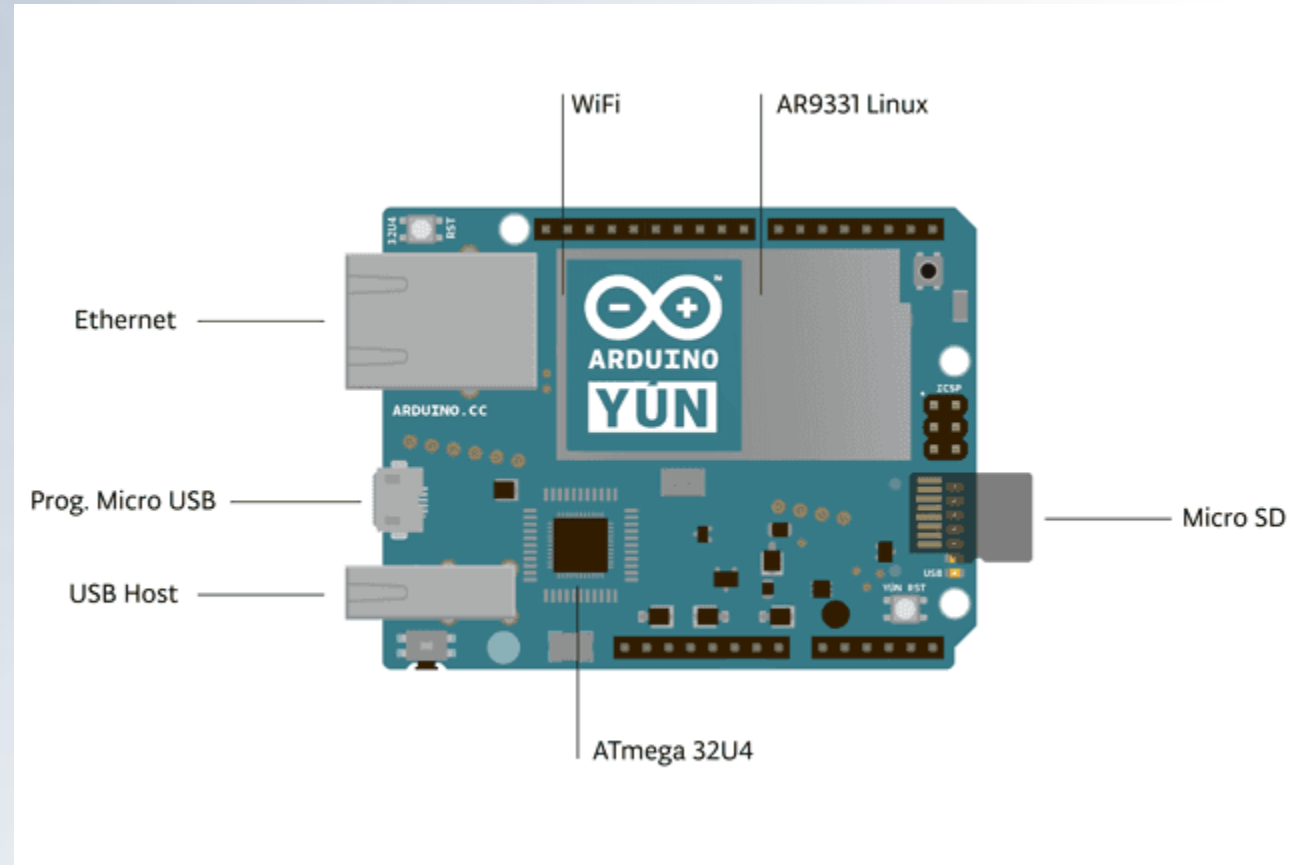


Popular IoT Development Boards

1. Arduino Yún
2. Microduino
3. Intel Galileo
4. Intel Edison
5. Beagle Board
6. Raspberry Pi Wireless Inventors Kit (RasWIK)

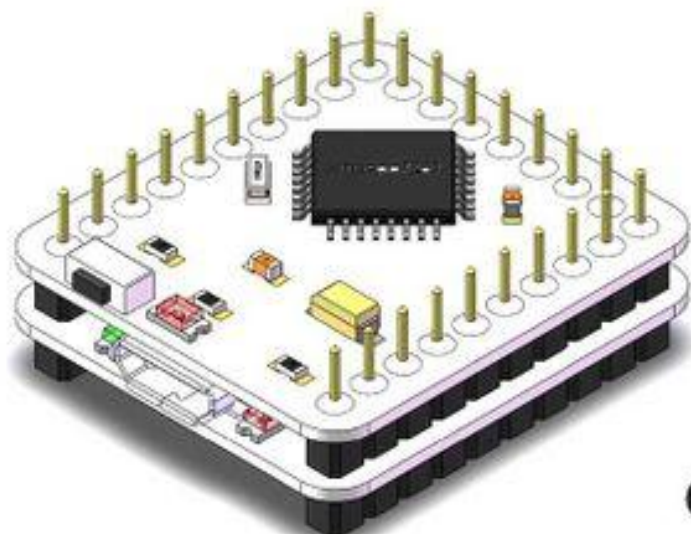
1. Arduino Yún

Arduino Yún board uses microcontroller ATmega32u4 that supports Arduino and includes Wi-Fi, Ethernet, USB port, micro-SD card slot and three reset buttons. The board also combines with Atheros AR9331 that runs Linux.



2. Microduino

- Microduino is a small board compatible with Arduino that can be stacked with the other boards.
- All the hardware designs are open source.

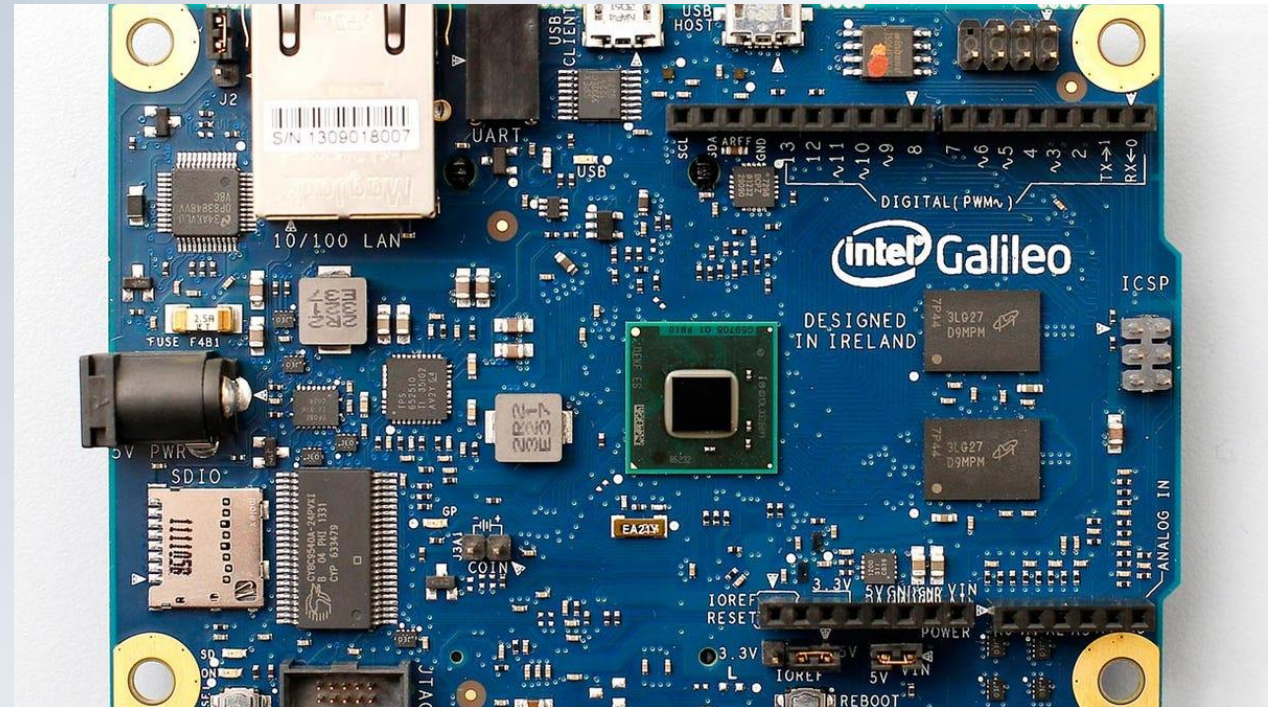


Small as a quarter
Stackable through UPin-27
Smart as Arduino
Open source hardware
Community for designers

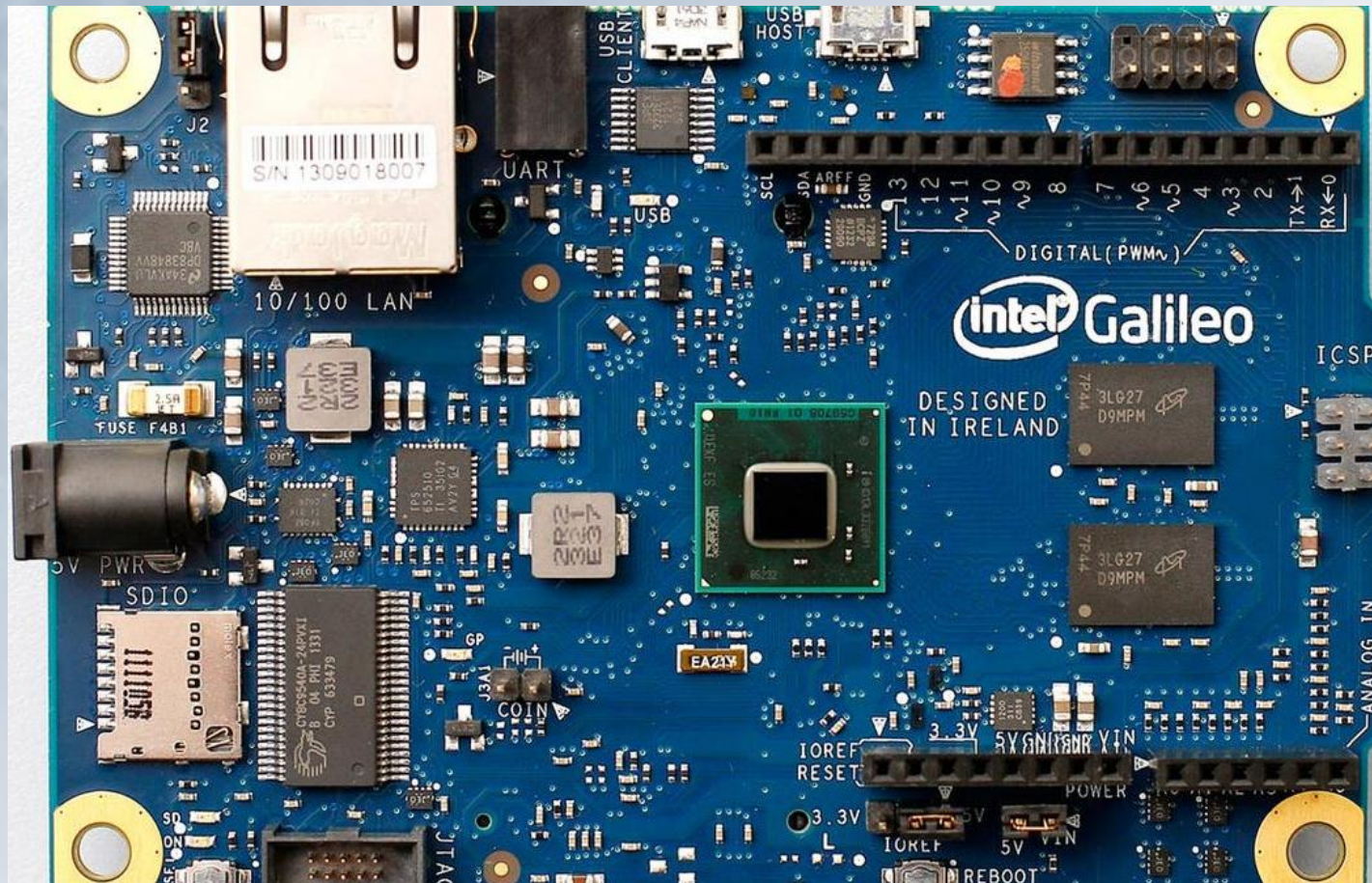
Microduino is a Product of Maker/Module 1.0 Core Module

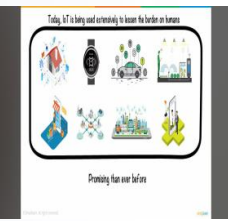
3. Intel Galileo

- Intel Galileo is a line of Arduino-certified development boards. Galileo is based on Intel x86 architecture. It is open-source hardware that features the Intel SOC X1000 Quark based Soc.
- Galileo is pin-compatible with Arduino. It has 20 digital I/O (12 GPIOs fully native), 12-bit PWM for more precise control, six analog inputs and supports power over Ethernet (PoE)



Intel Galileo





4. Intel Edison

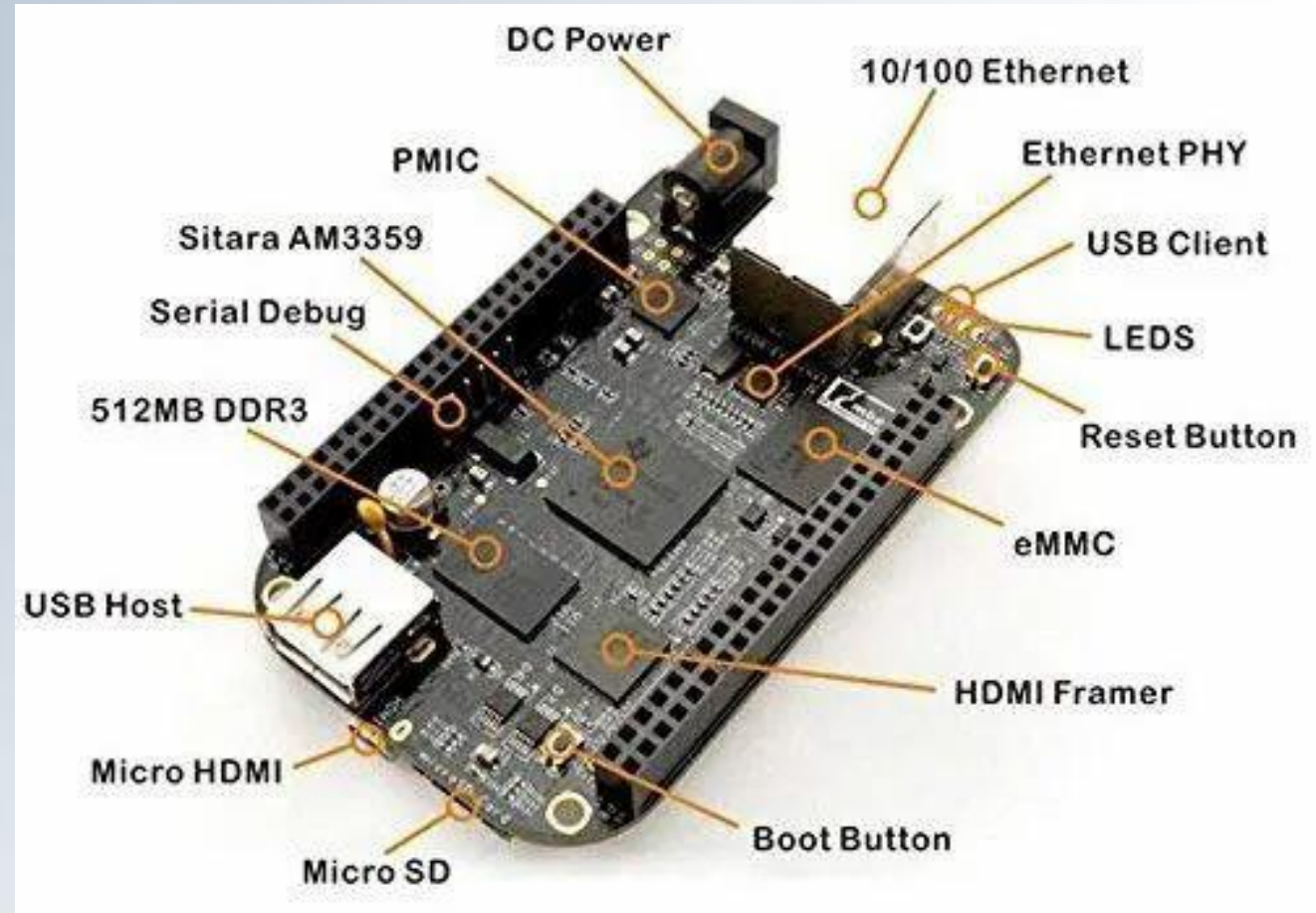
- Intel Edison19 is a compute module.
- It enables creation of prototypes and fast development of prototyping projects and rapidly produces IoT and wearable computing devices.
- It enables seamless device internetworking and device-to-cloud communication.
- It includes foundational tools. The tools collect, store and process data in the cloud, and process rules on the data stream.
- It generates triggers and alerts based on advanced analytics.

4. Intel Edison



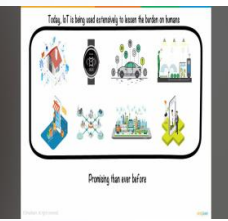
5. Beagle Board

- Beagle Bone based board has very low power requirement.
- It is a card-like computer which can run Android and Linux.
- Both the hardware designs and the software for the IoT devices are open source.



6. Raspberry Pi Wireless Inventors Kit (RasWIK)

- RasWIK enables Raspberry Pi Wi-Fi connected devices.
- It includes documentation for 29 different projects or you can come up with one of your own.
- There is a fee for the devices but all of the included code is open source, and you can use it to build commercial products as well.



1.5.2 Role of RFID and IoT Applications

- Earlier IoT systems were internet-connected RFID based systems.
- RFID enables tracking and inventory control, identification in supply chain systems, access to buildings and road tolls or secured store centre entries, and devices such as RFID-based temperature sensors.
- RFID networks have new applications in factory design, 3PL-management, brand protection, and anti-counterfeiting in new business processes for payment, leasing, insurance and quality management.



1.5.3 Wireless Sensor Networks (WSNs)

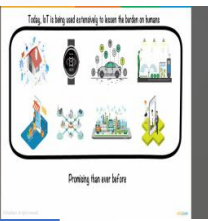
- Sensors can be networked using wireless technology and can cooperatively monitor physical or environmental conditions.
- Sensors acquire data from remote locations, which may not be easily accessible.
- Each wireless sensor also has communication abilities for which it uses a radio-frequency transceiver.
- Each node either has an analog sensor with signal conditioner circuit or a digital sensor.
- Sensing can be done to monitor temperature, light intensity, presence of darkness, metal proximity, traffic, physical, chemical and biological data etc.



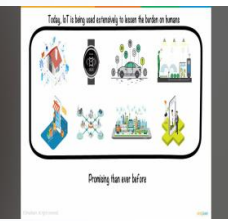
Wireless Sensor Network (WSN)

- Wireless Sensor Network (WSN) is defined as a network in which each sensor node connects wirelessly and has capabilities of computations for data compaction, aggregation and analysis plus communication and networking. WSN node is autonomous.
- Autonomous refers to independent computing power and capability to send requests and receive responses, and data forward and routing capabilities.

WSN Node



- A WSN node has limited computing power. It may change topology rapidly.
- The WSN network in the topology-changing environment functions as an ad-hoc network.
- A WSN network in that environment is generally self-configuring, self-organising, self-healing and self-discovering.



1.6 M2M COMMUNICATION

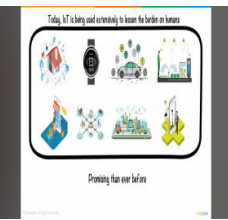
- Machine-to-machine (M2M) refers to the process of communication of a physical object or device at machine with others of the same type, mostly for monitoring but also for control purposes. Each machine in an M2M system embeds a smart device.
- The device senses the data or status of the machine, and performs the computation and communication functions M2M technology involves the automatic and streamlined sharing of information between two or more separate devices.
- Common examples include smart home meters, vehicle telemetry services, asset tracking, wearable technologies, and automated supply chain management (SCM).

Difference between IoT and M2M

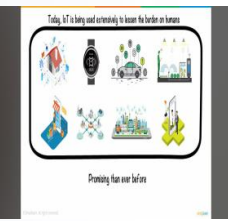


Basis of	IoT	M2M
Abbreviation	Internet of Things	Machine to Machine
Intelligence	Devices have objects that are responsible for decision making	Some degree of intelligence is observed in this.
Connection type used	The connection is via Network and using various communication types.	The connection is a point to point
Communication protocol used	Internet protocols are used such as HTTP, FTP, and Telnet.	Traditional protocols and communication technology techniques are used
Data Sharing	Data is shared between other applications that are used to improve the end-user experience.	Data is shared with only the communicating parties.
Internet	Internet connection is required for communication	Devices are not dependent on the Internet.
Type of Communication	It supports cloud communication	It supports point-to-point communication.
Components	Devices/sensors, connectivity, data processing, user interface	Device, area networks, gateway, Application server.
Examples	Smart wearables, Big Data and Cloud, etc.	Sensors, Data and Information, etc.

Difference between IoT and M2M



Basis of	IoT	M2M
Computer System	Involves the usage of both Hardware and Software.	Mostly hardware-based technology
Scope	A large number of devices yet scope is large.	Limited Scope for devices.
Business Type used	Business 2 Business(B2B) and Business 2 Consumer(B2C)	Business 2 Business (B2B)
Open API support	Supports Open API integrations.	There is no support for Open APIs
It requires	Generic commodity devices.	Specialized device solutions.
Centric	Information and service centric	Communication and device centric.
Approach used	Horizontal enabler approach	Vertical system solution approach .



Features of M2M Communication

- **Efficient Energy Use for Enhanced M2M:** The M2M system conserves energy, leading to improved performance in M2M applications.
- **Seamless Data Exchange in M2M:** Network operators utilize organized data packets to ensure smooth information sharing among machines in M2M communication.
- **Rapid Event Detection:** Through monitoring, the system swiftly identifies events.
- **Flexible Data Timing:** Data transfers can tolerate minor delays.
- **Scheduled Information Sharing:** Data is transmitted or received at specific, pre-defined times.
- **Location-Based Device Notifications:** Devices receive alerts when entering specific areas.
- **Steady and Small-Scale Data Transfer:** The system maintains a consistent flow of small data packets.

1.6.1 M2M to IoT

- IoT technology in industry involves the integration of complex physical machinery M2M communication with the networks of sensors, and uses analytics, machine learning, and knowledge discovery software.
- M2M technology closely relates to IoT when the smart devices or machines collect data which is transmitted via the Internet to other devices or machines located remotely.
- The close difference between M2M and IoT is that M2M must deploy device to device, and carry out the coordination, monitoring, controlling of the devices and communicate without the usage of Internet whereas IoT deploys the internet, server, internet protocols and server or cloud end applications, services or processes.

1.6.2 M2M Architecture

M2M architecture consists of three domains (Figure 1.9):

1. M2M device domain - consists of three entities: physical devices, communication interface and gateway. Communication interface is a port or a subsystem, which receives the input from one end and sends the data received to another.
2. M2M network domain - consists of M2M server, device identity management, data analytics and device management similar to IoT architecture (connect + collect + assemble + analyse) level.
3. M2M application domain - consists of application for services, monitoring, analysis and controlling of devices networks.

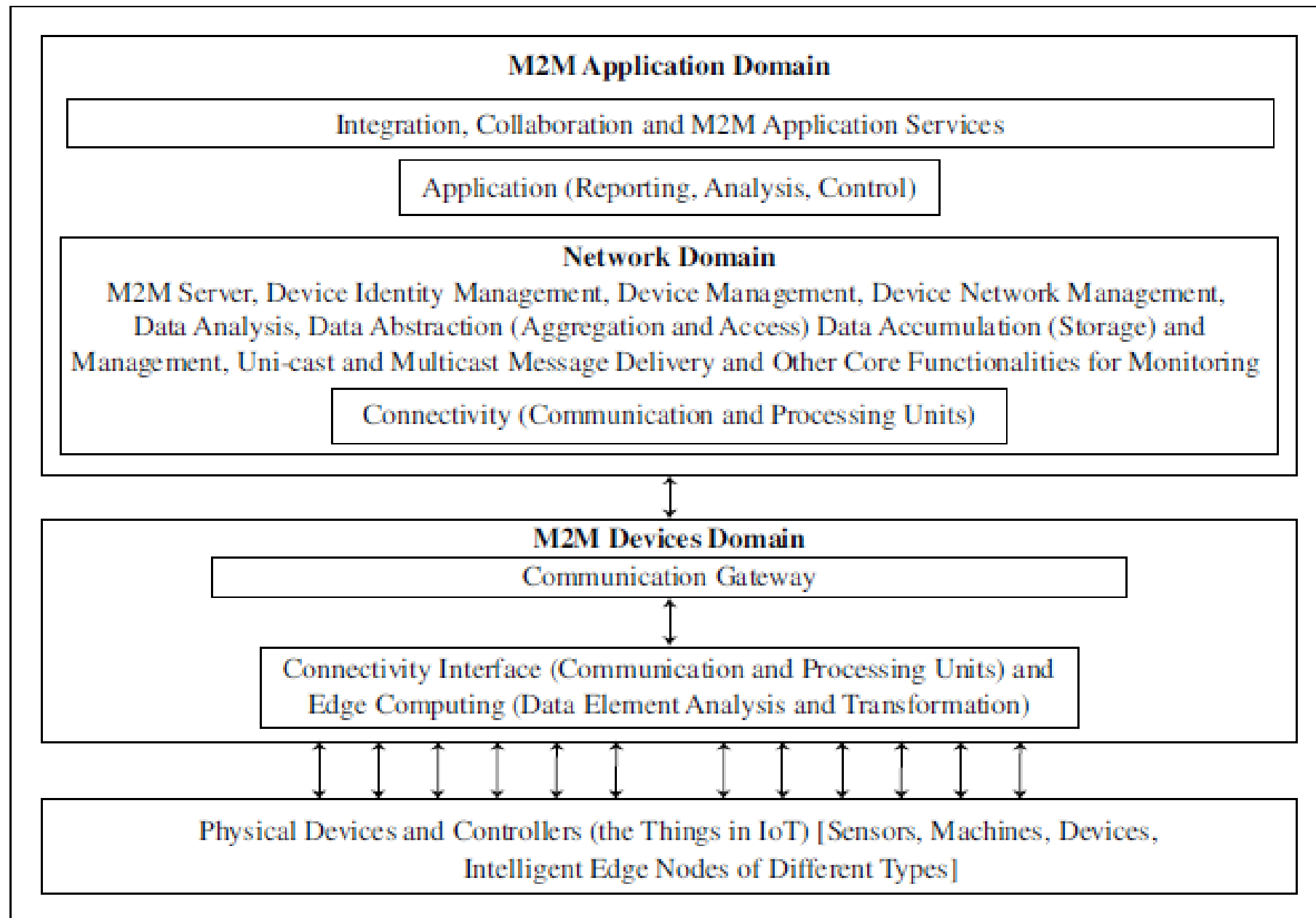
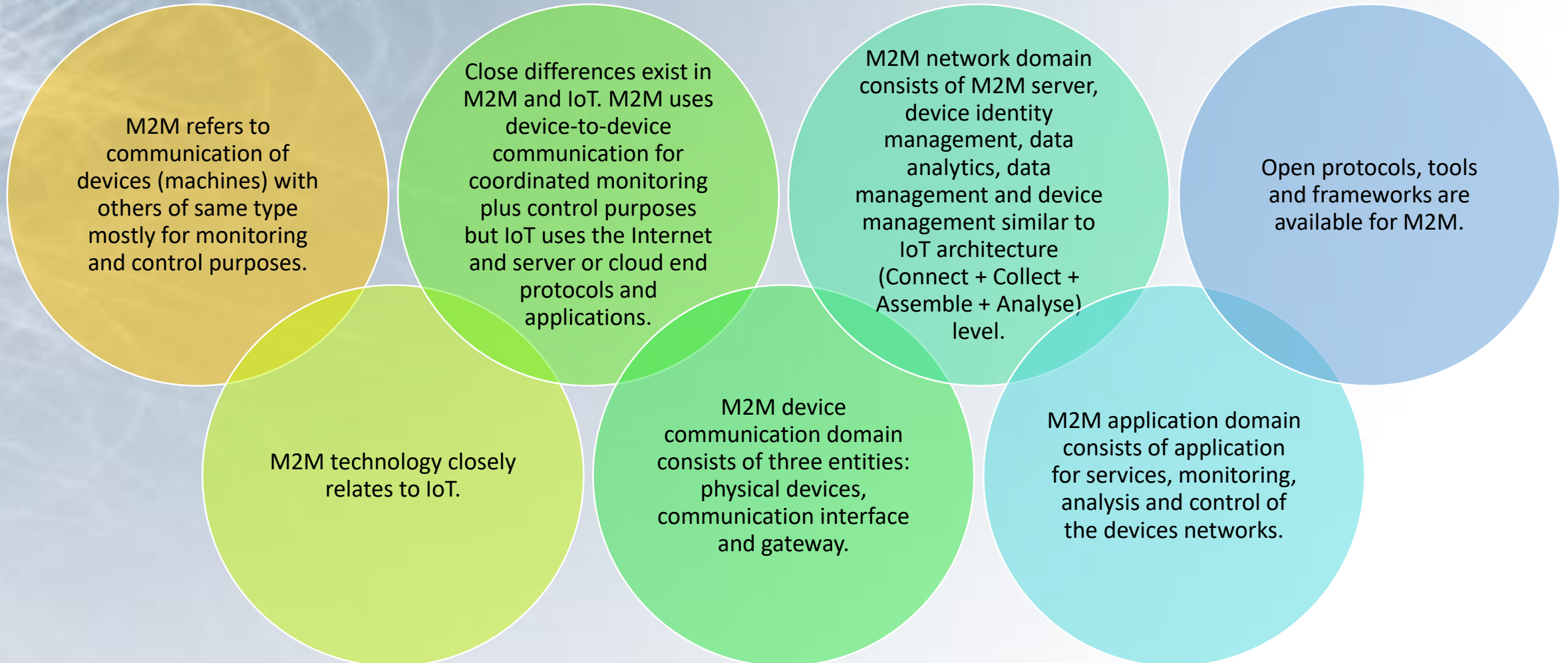


Figure 1.9 Three domains of M2M architecture

Summary



1.7 EXAMPLES OF IoT

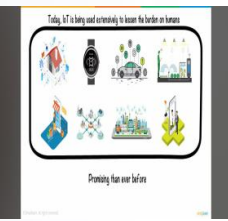
- 1.7.1 Wearable Smart Watch
- 1.7.2 Smart Home
- 1.7.3 Smart Cities

[Reference: Raj Kamal “INTERNET OF THINGS”, McGraw-Hill, 1ST Edition, 2016]

Design Principles for Connected Devices

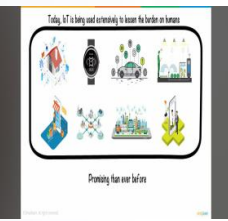
Unit 1
Chapter 2





Introduction

- When a letter is written then it is written according to a protocol (etiquette).
- To send a letter, it is first put in an envelope, and then the envelope is marked with the receiver's address at the centre, sender's address at left hand bottom area, stamp(s) is/are affixed at right hand top corner and the type of post is mentioned on top line in the centre.
- All letters are then gathered (stacked) and the stack is sent to the target city.
- Each action takes place according to a specified protocol at each stage (layer).
- Similarly, when data is transferred from a sensor, then functional units create a stack for data communication to an application or service.



What is Data?

- IoT or M2M device data refers to the data meant for communication to an application, service or process.
- Data also refers to data received by a device for its monitoring or for actions at actuator in it.
- **Data stack:** denotes the data received after the actions at various in-between layers (or levels or domains).
- **Layers in Open Systems Interconnection (OSI) model** are Application, Presentation, Session, Transport, Network, Data-link and Physical.

Key terms to know..

Following are the key terms which need to be understood to learn the design principles of connected devices for IoTs:

1. Layer
2. Physical Layer
3. Application Layer
4. Level
5. Domain
6. Gateway
7. IP
8. Header
9. Packet
10. Protocol Data Unit (PDU)
11. Maximum Transmission Unit (MTU)
12. Star network
13. Mesh network End-point device or node
14. Coordinator
15. Master
16. Slave
17. Router
18. ISM band
19. Service
20. Process

2.2 IoT/M2M SYSTEMS, LAYERS AND DESIGNS STANDARDISATION

A number of international organizations have taken action for IoT design standardization. Following are the examples:

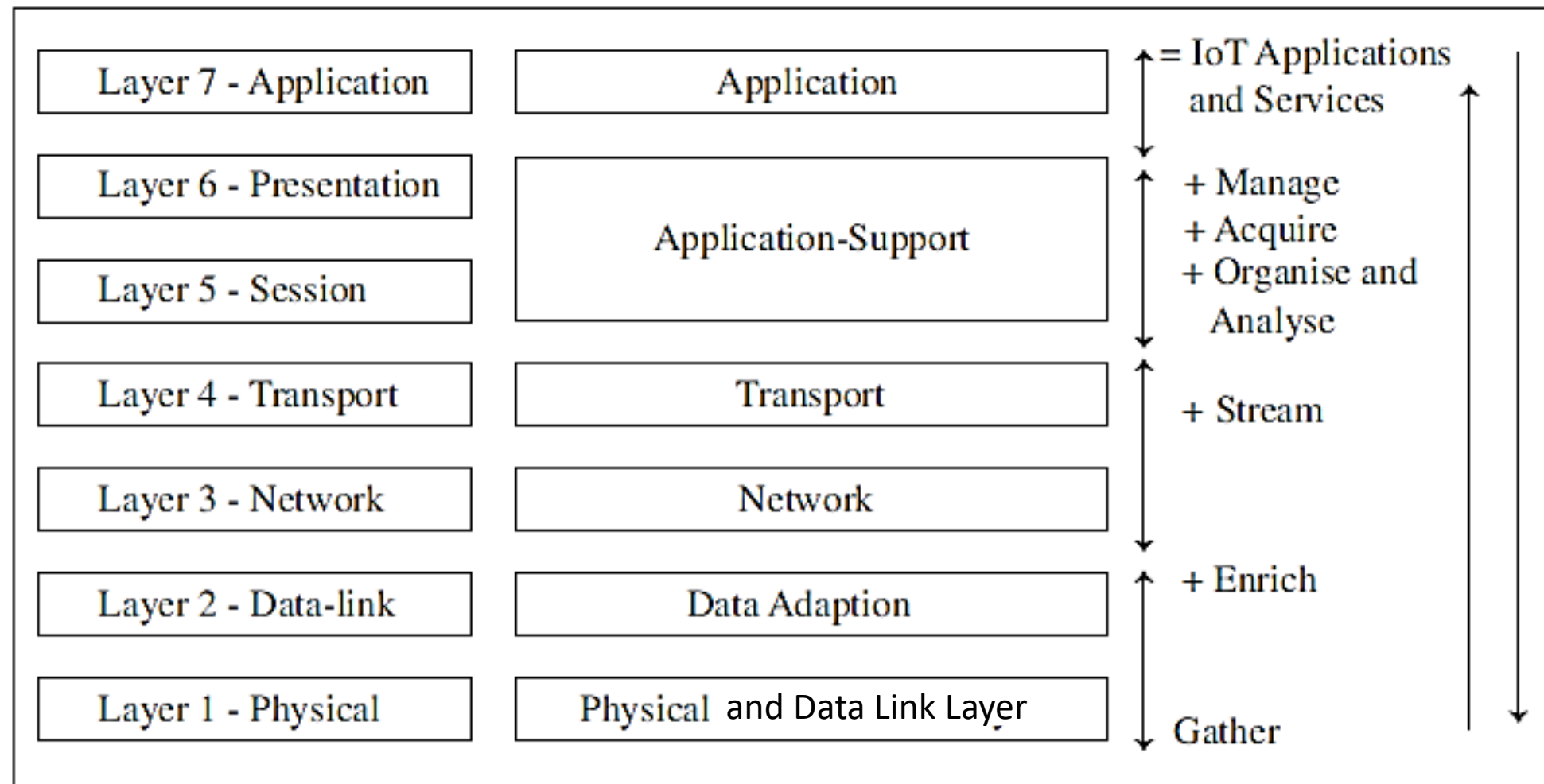
1. Internet Engineering Task Force (IETF)
2. International Telecommunication Union for Telecommunication (ITU-T)
3. European Telecommunication Standards Institute (ETSI)
4. Open Geospatial Consortium (OGC)

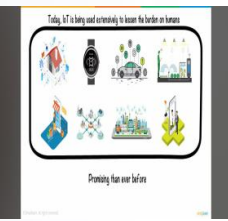


2.2.1 Modified OSI Model for the IoT/M2M Systems

- OSI protocols mean a family of information exchange standards developed jointly by the ISO and the ITU-T.
- The seven-layer OSI model is a standard model.
- It gives the basic outline for designing a communication network.
- Various models for data interchanges consider the layers specified by the OSI model, and modify it for simplicity according to the requirement.
- Similarly, IETF suggests modifications in the OSI model for the IoT/M2M.

- Figure 2.1 shows a classical seven-layer OSI model (on the left) and the modifications in that model proposed by IETF (in the middle).



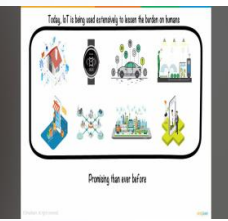


- Figure 2.1 shows a classical seven-layer OSI model (on the left) and the modifications in that model proposed by IETF (in the middle).
- Data communicates from device end to application end.
- Each layer processes the received data and creates a new data stack which transfers it to the next layer.
- The processing takes place at the in-between layers, i.e. between the bottom functional-layer to the top layer.
- Device end also receives data from an application/service after processing at the in-between layers.
- Figure 2.1 also shows a similarity with the conceptual framework in Equation 1.2:

Gather + Enrich + Stream + (Manage + Acquire + Organise and Analyse) = IoT Applications and Services

Case study

- What are the architectural layers in a modified OSI model for Internet of smart streetlights application in the model for Internet of streetlights(Example 1.1)



Problem

What are the architectural layers in a modified OSI model for Internet of smart streetlights application in Example 1.1?

Solution

Consider a model for Internet of streetlights (Figure 1.1). Following are the layers for data interchange in the modified OSI model:

- L1: It consists of smart sensing and data-link circuits with each streetlight transferring the sensed data to L2.
- L2: It consists of a group-controller which receives data of each group through Bluetooth or ZigBee, aggregates and compacts the data for communication to the Internet, and controls the group streetlights as per the program commands from a central station.
- L3: It communicates a network stream on the Internet to the next layer.
- L4: The transport layer does device identity management, identity registry and data routing to the next layer
- L5: The application-support layer does data managing, acquiring, organising and analysing, and functionalities of standard protocols such as CoAP, UDP and IP.
- L6: The application layer enables remote programming and issue of central station directions to switch on-off and commands of services to the controllers along with monitoring each group of streetlights in the whole city.

2.2.2 ITU-T Reference Model

- Figure 2.2 shows the ITU-T reference model RM1. It also shows correspondence of the model with the six-layers modified OSI model (Figure 2.1). The figure also shows a comparison with CISCO IoT reference model RM2 (Figure 1.4). RM1 considers four layers which are:
 1. Lowest layer, L1, is the device layer and has device and gateway capabilities.
 2. Next layer, L2, has transport and network capabilities.
 3. Next layer, L3, is the services and application-support layer. The support layer has two types of capabilities—generic and specific service or application-support capabilities.
 4. Top layer, L4, is for applications and services.

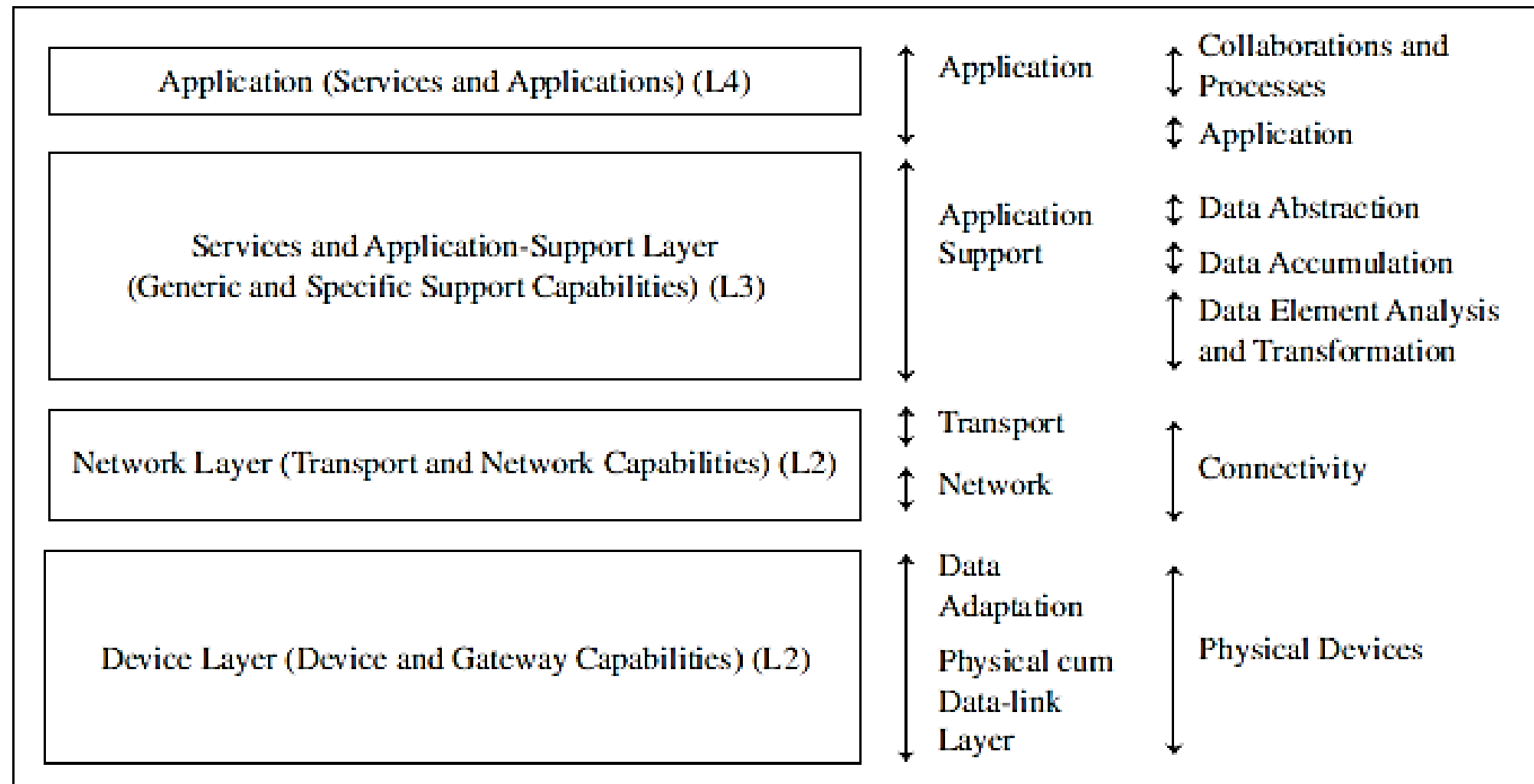


Figure 2.2 ITU-T reference model RM1, its correspondence with six layers of modified OSI and a comparison with seven levels suggested in CISCO IoT reference model RM2

ITU-T four layers

ITU-T recommends four layers, each with different capabilities. A comparison of ITU-T RM1 with the six-layer OSI model can be made as follows:

- RM1 device layer capabilities are similar to data-adaptation and physical cum datalink layers.
- RM1 network layer capabilities are similar to transport and network layers.
- RM1 upper two layer capabilities are similar to top two layers.

Comparison between RM1 and RM2

A comparison of ITU-T RM1 with the CISCO IoT reference model (RM2) can be made as follows:

- RM1 L4 capabilities are similar to RM2 collaborations and processes, and application top two levels.
- RM1 L3 capabilities are similar to RM2 three middle-level functions of data abstraction accumulation, analysis and transformation.
- RM1 L2 layer capabilities are similar to RM2 functions at connectivity level.
- RM1 L1 device layer capabilities are similar to RM2 functions at physical devices level.

2.2.3 ETSI M2M Domains and High-level Capabilities

- A domain specifies the functional areas. High-level architecture means architecture for functional and structural views.

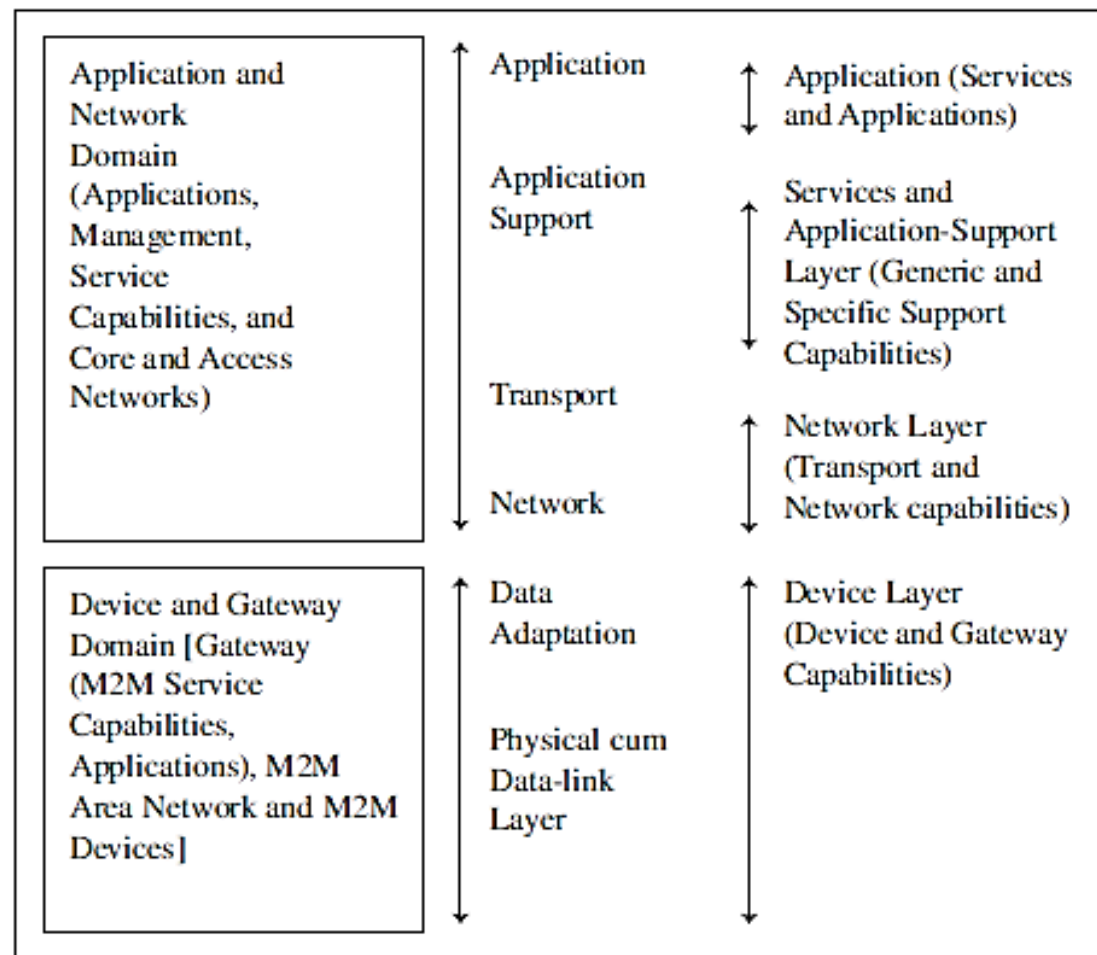


Figure 2.3 ETSI M2M domain architecture and its high-level capabilities, and its correspondences with six layers of modified OSI and four layers of ITU-T reference model

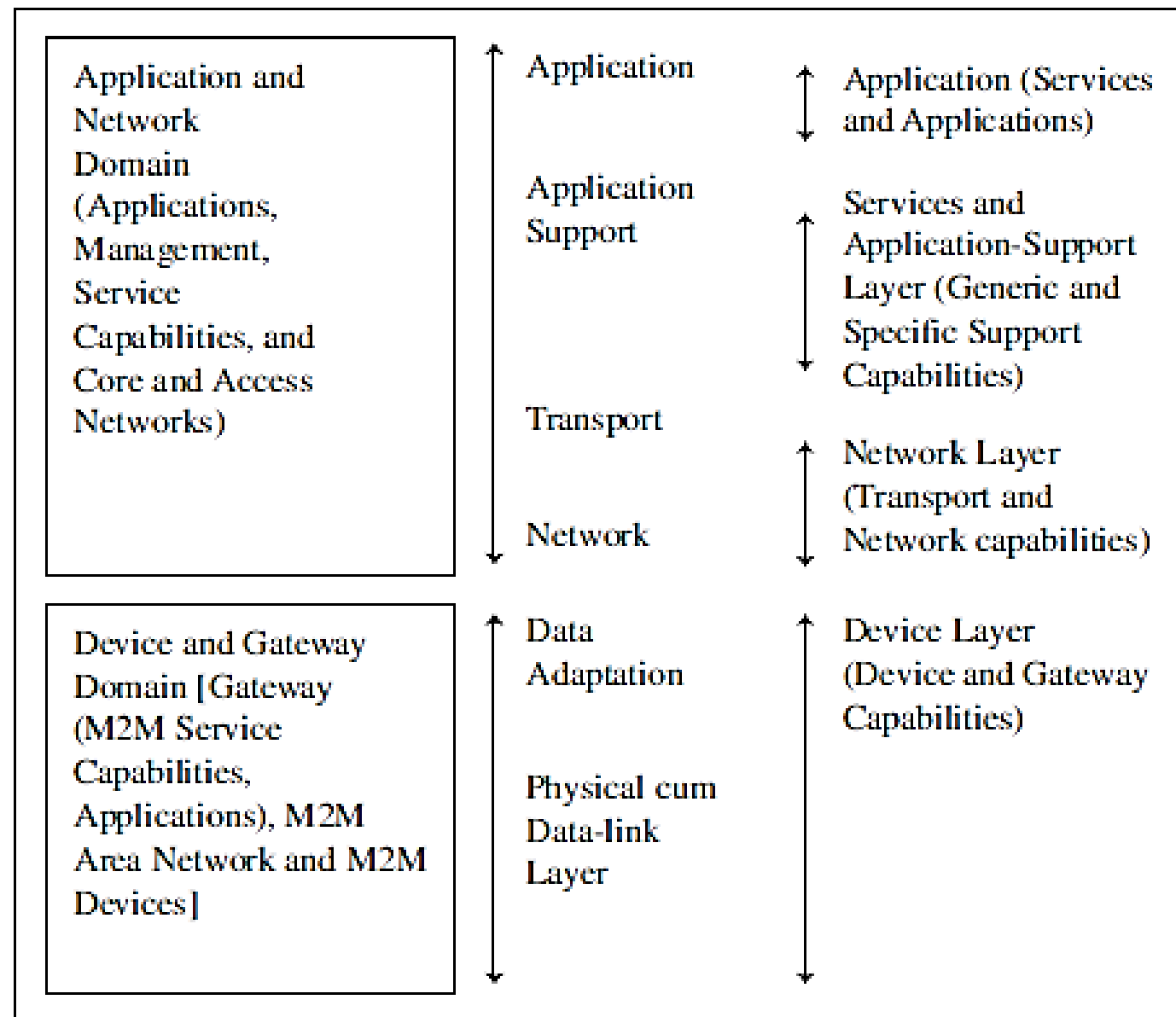


Figure 2.3 ETSI M2M domain architecture and its high-level capabilities, and its correspondences with six layers of modified OSI and four layers of ITU-T reference model

Case Study : What are domains and their service capabilities in ETSI high-level architecture for applications and services in Internet of ATM machines?



Solution

ETSI high-level architecture for applications and services in Internet of ATM machines has two domains:

Device and gateway domain:

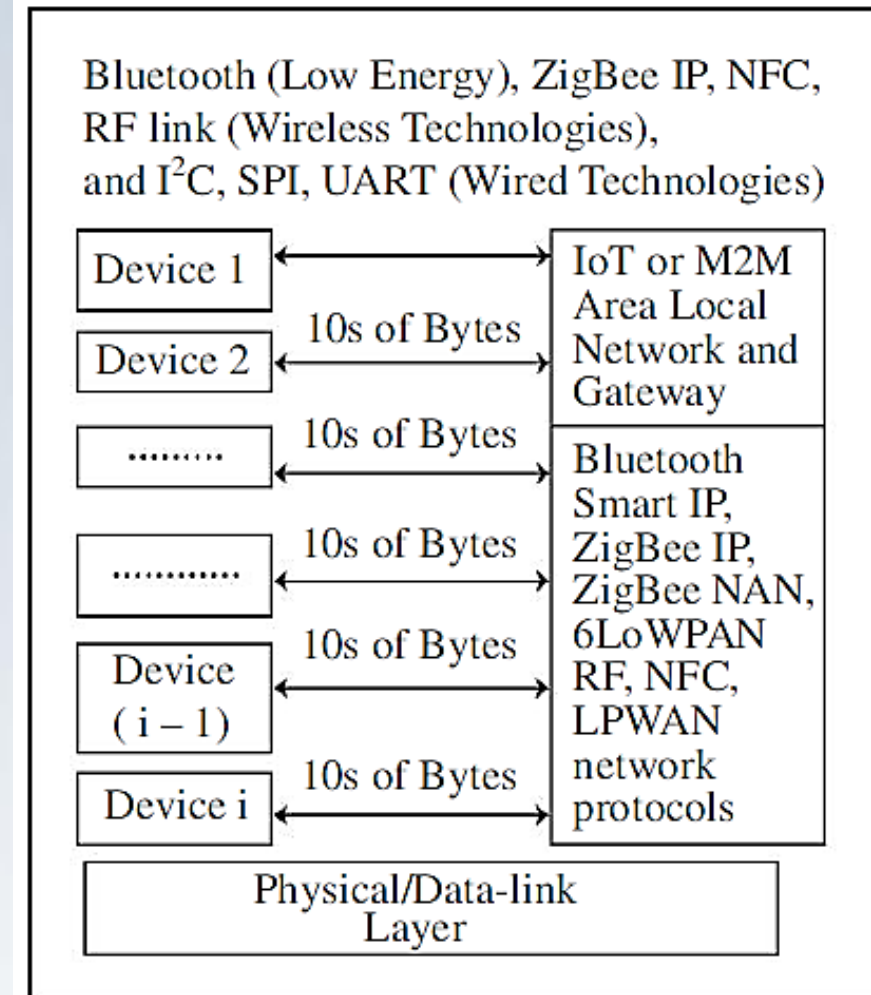
- Device refer to cards and ATMs, while ATM service capabilities and ATM applications are present at the ATM gateway. The gateway has system for acquiring the card as well as banking data. Data interchange between an ATM machine and the bank server takes place through the gateway.
- The domain has cash dispensing and surveillance systems.
- All the ATM systems network through an access network. The gateway communicates the data after enriching and transcoding according to the network protocol between an AP and data for the machine.
- A domain subsystem monitors cash dispensing and other services.

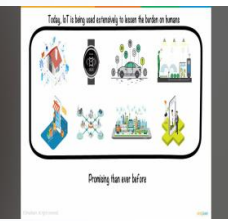
Applications and network domain:

- Application and network domain has two functional units—ATM management functions and network management functions. It has banking applications and service capabilities for the ATMs.
- It communicates with the bank CoRE network which connects all the access networks of ATM gateways.

2.3 COMMUNICATION TECHNOLOGIES

- Physical cum data-link layer in the model consists of a LAN/PAN.
- A local network of IoT or M2M device deploys one of the two types of technologies - wireless or wired communication technologies.
- Figure 2.4 Connected devices 1st to ith connected to the local network and gateway using the WPAN or LPWAN network protocols





- **2.3.1 Wireless Communication Technology**

- Near-Field Communication
- Bluetooth BR/EDR and Bluetooth Low Energy
- ZigBee IP/ZigBee SE 2.0
- Wi-Fi
- RF Transceivers and RF Modules
- GPRS/GSM Cellular Networks-Mobile Internet
- Wireless USB

- **2.3.2 Wired Communication Technology**

- UART/USART Serial Communication
- Serial Peripheral Interface
- I2C Bus
- Wired USB
- Ethernet

2.3.3 Communication Technologies— A Comparison

Property	NFC	BT LE	ZigBee IP	WLAN 802.11
IEEE Protocol		802.15.1	802.15.4	802.11z
Physical Layer	848, 424, 212, 106 kbps	2.4 GHz (LE-DSSS)	2.4 GHz or 915 MHz, 868 MHz and 433 MHz DSSS MAC layer CSMA/CA	2.4 GHz Two PHY layers MAC layer CSMA/CD
Data Transfer Rate	106 kbps	1 Mbps	250 kbps (2.4 GHz, 40 kbps 915 MHz, 20 kbps 868.3 MHz)	11 Mbps/54 Mbps
Form Factor and Range	10–20 cm	Small	Small 10 m to 200 m	Bigger
Protocol Stack		Small in LE	127 B	Bigger than WPAN devices

Property	NFC	BT LE	ZigBee IP	WLAN 802.11
Power Dissipation	Very low	Lower than ZigBee, much lower than WLAN 802.11	2 mW Router and 0.1 mW for end-device Much lower than WLAN 802.11	Much Higher than ZigBee
Set up/ Connection/ Disconnection Intervals	0.1 s	3s Connection time < 3 ms	20 ms Connection time < 10 ms	
Security	—	AES-CCM-128	AES-CCM-128	WEP
Applications	Payment wallet, short distance communication	WPAN, IoT/M2M devices, widely present in mobiles and tablets and, need addition circuit in sensors, actuators, controllers and IoT devices	WPAN, wider presence in sensors, actuators, controllers, automobile and medical electronic and IoT devices connectivity using IPv6, 6LoWPAN, ROLL, RPL and TLSv1.2	WLAN and WWAN network tablet, desktops, mobiles, devices with PCMCIA interface, home networking, Easy IPv4 connectivity

Property	NFC	BT LE	ZigBee IP	WLAN 802.11
Network	Point to point between active and passive devices	Star topology, peer-to-peer piconet expended by inter-piconets data transactions and synchronisation	Low power, mesh or peer-to peer star networks using end devices, coordinator, router, ZigBee IP border router	LAN topology IBSS, BSS and distributed BSSs for WWLAN widely used for Internet connectivity of mobiles, tablets, desktops
Network Characteristics	P2P mode, card emulation mode and reader mode Passive neighbour activation	Self-configuring, self-healing, self-discovery	Self-configuring, self-healing, self-discovery	Scalable, interoperability, security, integrity and reliability
Broadcast/ Multicast/Unicast	Unicast	Unicast	Unicast/ multicast	Unicast

2.4 DATA ENRICHMENT, DATA CONSOLIDATION AND DEVICE MANAGEMENT AT GATEWAY

- A gateway at a data-adaptation layer has several functions.
- These are *data privacy, data security, data enrichment, data consolidation, transformation* and *device management*.
- A gateway consists of the data enrichment, consolidation and IoT communication frameworks.
- The communication gateway enables the devices to communicate and network with the web.
- The communication gateway (Section 3.3.1) uses message transport protocols and web communication protocols for the Internet.
- The gateway includes two functions viz. data management and consolidation, and connected device management.

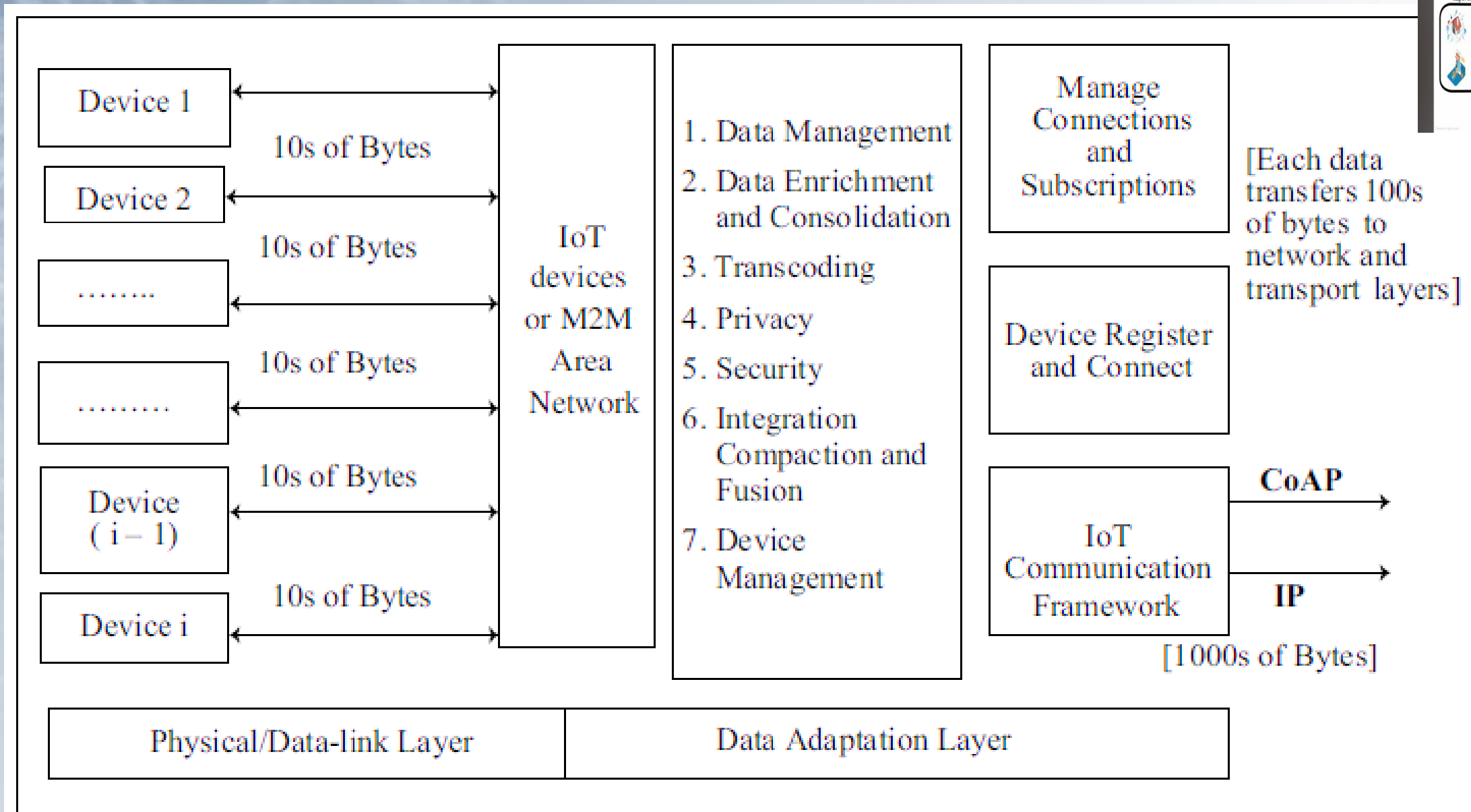


Figure 2.7 IoT or M2M gateway consisting of data enrichment and consolidation, device management and communication frameworks at the adaptation layer

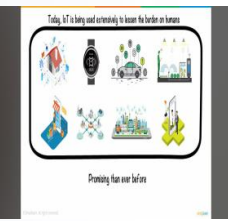
2.4.1 Data Management and Consolidation Gateway

- Gateway includes the provisions for one or more of the following functions: transcoding and data management.
- Following are data management and consolidation functions:
 - Transcoding
 - Privacy, security
 - Integration
 - Compaction and fusion



Transcoding

- The gateway renders the web response and messages in formats and representations required and acceptable at an IoT device.
- Similarly, the IoT device requests are adapted, converted and changed into required formats acceptable at the server by the transcoding software.
- In multimedia data transfer, transcoding **converts formats, data, and code from server to client devices** like mobile TVs, Internet TVs, VoIP phones, or smartphones. Transcoding applications may also involve filtering, compression, or decompression.
- Transcoding proxies can operate on client systems or application servers, providing conversion, computation, and analysis capabilities, while gateways offer conversion and computational capabilities only.

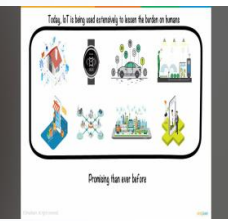


Privacy

- Privacy is crucial for protecting sensitive data such as patient medical records and inventory information during transfers between locations within a company. When designing applications, it's essential to prioritize privacy considerations, ensuring data remains anonymous to unauthorized recipients.
- Key components of a privacy model include:
 - Device and application identity management
 - Authentication
 - Authorization
 - Trust
 - Reputation
- Encrypting data sources and managing device IDs help enforce privacy. Analyzed decrypted data should only be accessible to authorized applications or processes. In the context of IoT or M2M data, ensuring data access is restricted to intended beneficiaries is crucial.
- During data transfer, it's vital to prevent future misuse by stakeholders. Establishing and maintaining trust and reputation are essential components of both static and dynamic relationships within the privacy framework.

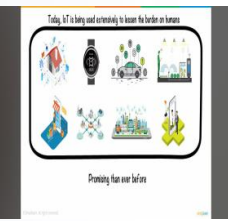
Secure Data Access

- Access to data needs to be secure.
- The design ensures the authentication of a request for data and authorization for accessing a response or service.
- It may also include auditing of requests and accesses of the responses for accountability in future.
- End-to-end security is another aspect while implies using a security protocol at each layer, physical, logical link and transport layers during communication at both ends in a network.



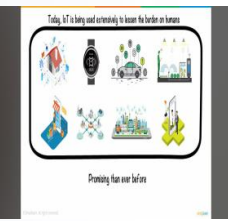
Data Gathering and Enrichment

- IoT/M2M applications involve actions such as data-gathering (acquisition), validation, storage, processing, reminiscence (retention) and analysis.
- *Data gathering* refers to data acquisition from the devices/devices network. Four modes of gathering data are:
 1. Polling
 2. Event-based gathering
 3. Scheduled interval
 4. Continuous monitoring
- *Data enrichment* refers to adding value, security and usability of the data.



Data Gathering and Enrichment

1. Polling refers to the data sought from a device by addressing the device; for example, waste container filling information in a waste management system.
2. Event-based gathering refers to the data sought from the device on an event; for example, when the device reaches near an access point or a card reaches near the card reader or an initial data exchange for the setup of peer-to-peer or master-slave connection of BT device using NFC.
3. Scheduled interval refers to the data sought from a device at select intervals; for example, data for ambient light condition in Internet of streetlights.
4. Continuous monitoring refers to the data sought from a device continuously; for example, data for traffic presence in a particular street ambient light condition in Internet of streetlights

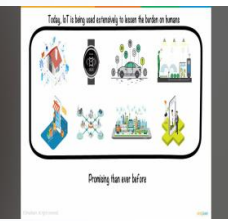


Energy Dissipation in Data Dissemination

- Energy consumption for data dissemination is an important consideration in many devices in WPANs and in wireless sensor nodes (WSNs). This is due to limited battery life.
- Energy is consumed when performing computations and transmissions.
- Higher the data rate, the greater will be the energy consumed.
- Higher is RF used, the greater will be the energy consumed.
- Higher the gathering interval, the lower will be the energy consumed.
- Energy efficient computations can be done by using concepts of data aggregation, compaction and fusion.
- Lesser the data bytes communication, greater the acquisition intervals, and lower the data rate for data transfer, lesser the energy dissipation.

Data Source and Data Destination

- ID: Each device and each device resource is assigned an ID for specifying the data of source and a separate ID for data destination.
- Address: Header fields add the destination address (for example, 48-bit MAC address at Link layer, 32-bit IPv4 address at IP network and 128-bit IPv6 address at IPv6 network) and may also add the port (for example, port 80 for HTTP application).



Data Characteristics, Formats and Structures

- **Data characteristics** can be in terms of:
 - temporal data (dependent on the time),
 - Spatial data (dependent on location),
 - real-time data (generated continuously and acquired continuously at the same pace),
 - real-world data (from physical world for example, traffic or streetlight, ambient condition)
 - proprietary data (copy right data reserved for distribution to authorised enterprises) and
 - big data (unstructured voluminous data)
- Data received from the devices, **formats** before transmission onto Internet. The format can be in XML, etc. A file can be MIME (Multipurpose Internet Mail Extensions) type for Internet. MIME allows email messages to include various types of content beyond plain text, making it versatile for multimedia communication.
- **Structure** implies the ways for arranging the data bytes in sequences with size limit defined by the Protocol Data Unit (PDU) for that layer.

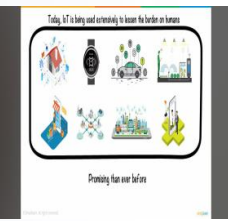
2.4.2 Device-management Gateway

- Device Management (DM) means provisioning for the device ID or address which is distinct from other resources, device activating, configuring (managing device parameters and settings), registering, deregistering, attaching and detaching.



2.5 EASE OF DESIGNING AND AFFORDABILITY

- Designing for IoT involves considering ease across physical, data-link, adaption, and gateway layers.
- It requires accessible low-cost hardware with open-source software, and simplified network setup.
- Design also encompasses RFID or card-based systems and affordable wireless sensors like Motes. A mobile terminal (Mote) which is a low cost device with an open-source OS (tiny OS) and software components.



2.5 EASE OF DESIGNING AND AFFORDABILITY (Continued)

- ZigBee IP or BT LE 4.2 are favored for smart environments – low cost.
- However, adding complexity to designs, such as programming umbrellas for scheduling SMS messages, may require additional instructions and manuals. Moreover, ensuring data transfer to trusted destinations using encryption tools can add complexity to connected device systems.



THANK YOU