INTERNET OF THINGS (IOT)

Asst. Prof. DR. MUHANED TH. M. AL-HASHIMI

Tikrit University

Collage Of Computer And Mathematical Science

2024 - 2025

PREDECESSORS OF IOT

LECTURE 3

2204 - 2025

25 Of February

Lecture outline

Predecessors of IoT

1. Introduction
2. Wireless Sensor Networks
3. Architectural components of WSN
4. Machine-to-Machine Communications
5. Architectural components of M2M
6. Differences between M2M and IoT
7. Cyber Physical Systems
8. Architectural components of CPS

Predecessors of IoT / Introduction

- Before delving into the details of the Internet of Things (IoT), a discussion on the base technologies, which make up the foundation of IoT, is required.
- A majority of these technologies, **before the IoT era**, were used separately for **sensing**, **decision-making**, and **automation tasks**.
- The range of **application domains** of these technologies **extended** from **regular** domains like healthcare, agriculture, home monitoring, and others to **specialized** domains such as military and mining.
- Some of these precursor technologies **still being used** and often **re-engineered** for IoT are wireless sensor networks (**WSN**), machine-to-machine (**M2M**) communications, and cyber physical systems (CPS).

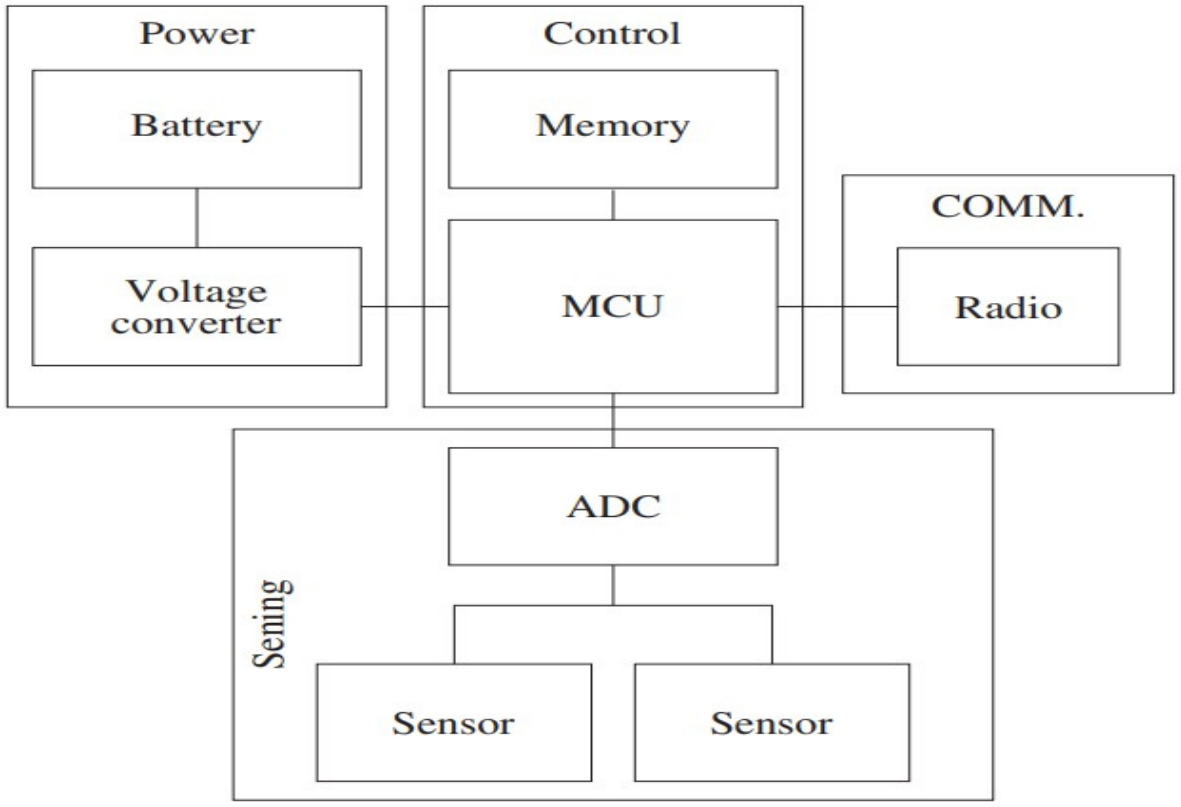
Predecessors of IoT/ wireless sensor networks (WSN)

- Wireless sensor networks (**WSN**), as the name suggests, is a **networking paradigm** that makes use of spatially distributed sensors for gathering information concerning the immediate environment of the sensors and collecting the information centrally.

- the sensors are not standalone devices but a combination of sensors, processors, and radio units referred to as **sensor nodes** sensing the environment and communicating the sensed data wirelessly to a remote location, which may or may not is connected to a backbone network.

Predecessors of IoT/ wireless sensor networks (WSN)

- Figure shows the block diagram of the various standard components of a typical WSN node.
- The exact specifications of each of these blocks vary depending on the implementation requirements and the network architect's choice.



The typical constituents of a WSN node

Predecessors of IoT/ wireless sensor networks (WSN)

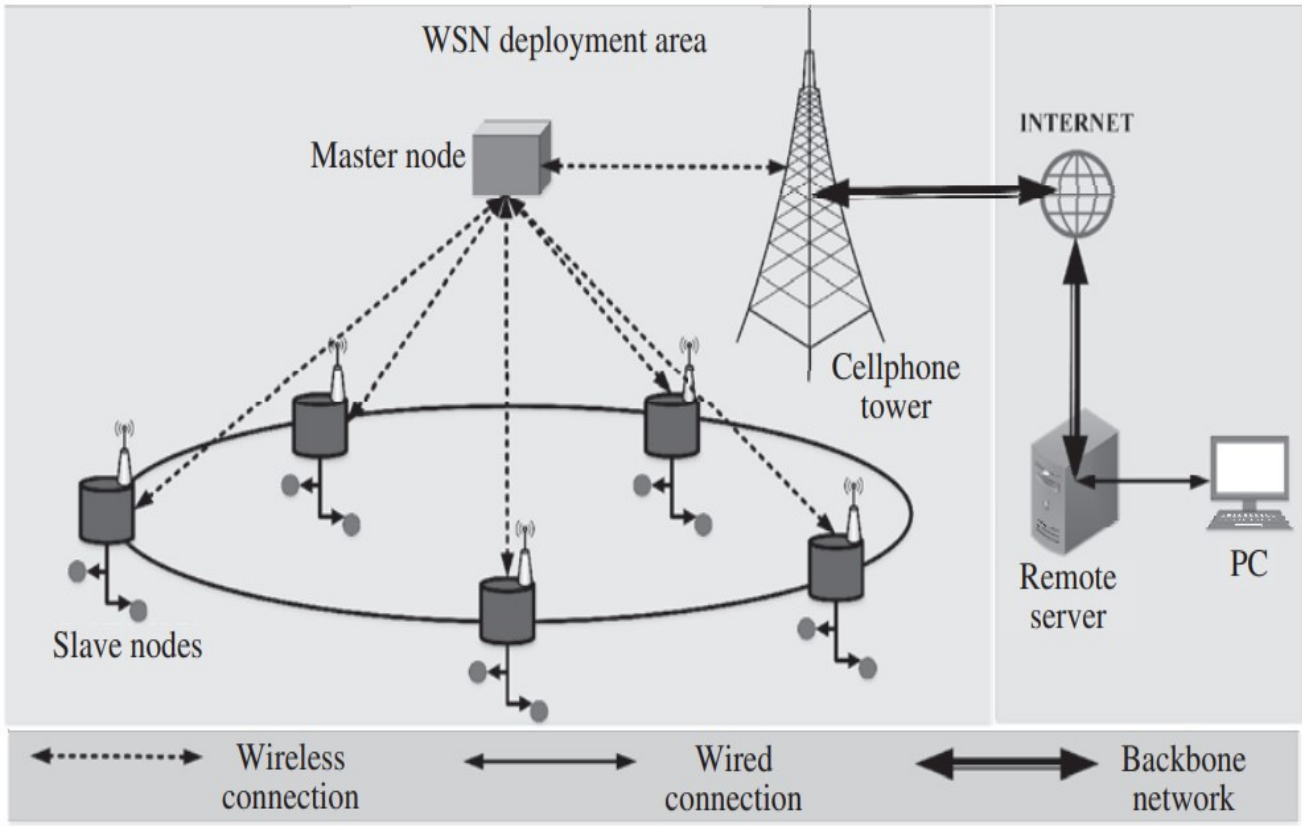
- Wireless sensor networks have found numerous applications in domains such as agriculture, healthcare, military, industries, mining, and others.

- The main **reason for the popularity of Wireless sensor networks** is attributed to the advantages they provide in the form of enhanced monitoring times, easy installation, and multiple implementations.

- **Implementations on a large scale** are possible due to high affordability, ease of replacement or upgradation, ease of modifying system parameters, ease of additional sensor integration with the sensor nodes, and other such **factors**.

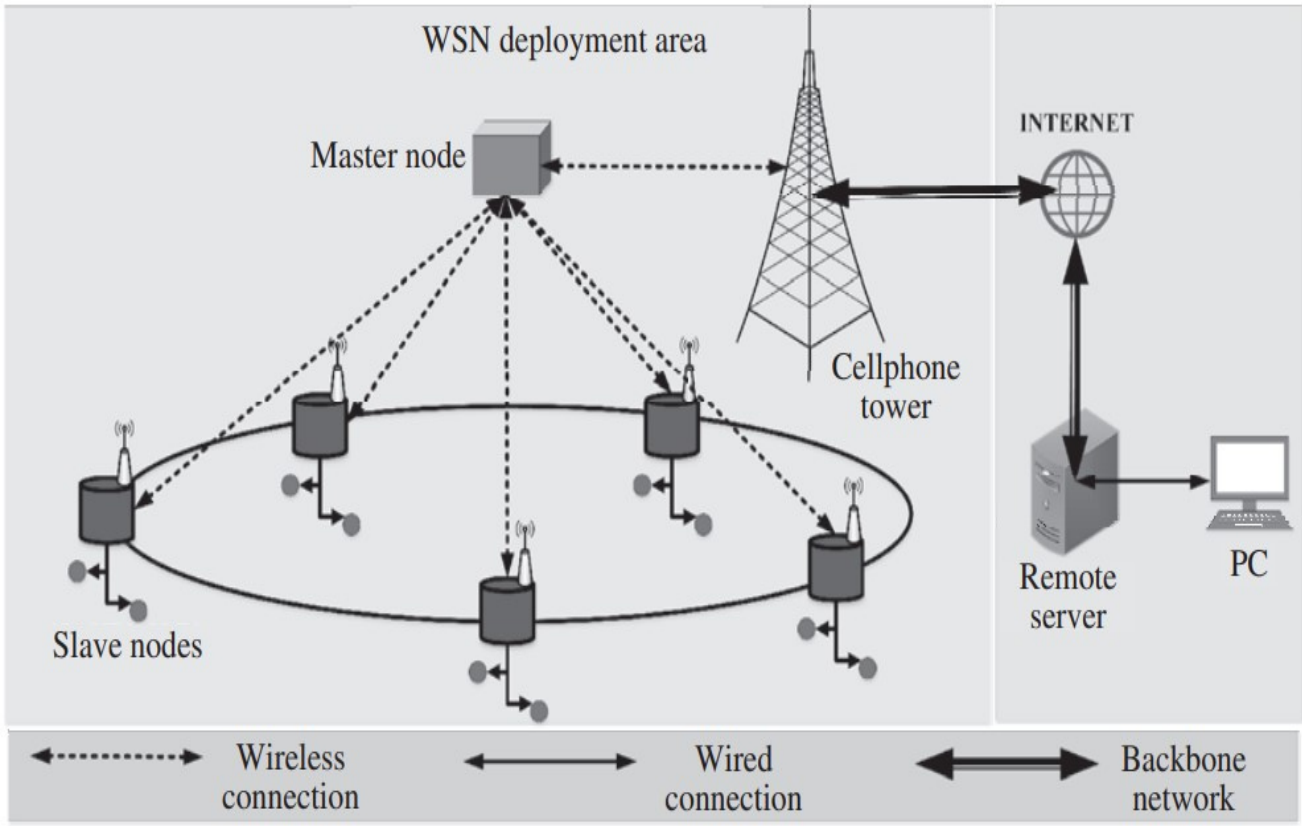
Predecessors of IoT/ wireless sensor networks (WSN)

- In a **master–slave** architecture, the **master**, is responsible for **collecting data** from various **sensor nodes** under its dominion or range of operations. The **sensor nodes** under the **range of the master node** are referred to as **slave nodes**.
- **Multiple** slave nodes **communicate** to the master node using **low-power short-range wireless radios** such as Zigbee, Bluetooth, and WiFi for **transferring** their **sensed data** to a remote central **server**.



Predecessors of IoT/ wireless sensor networks (WSN)

- Often, in popular WSN architectures, the
- **master node** connects the WSN to the Internet and acts as the gateway for the WSN.
- Upon **collecting data** from the slave nodes, the master node **pushes** the aggregated **data** to a **remotely** located central server using the **Internet**.
- The master node may be **linked** to the **Internet** through **cellular** connections, **another gateway**, or **directly** through a **backbone** infrastructure.



Predecessors of IoT/ wireless sensor networks (WSN)

- **WSNs must have the following distinguishing features:**

(i) **Fault Tolerance** تحمل الأخطاء : The occurrence of faults in WSN nodes should not take down the whole WSN implementation, or obstruction the transmission of data from non-faulty nodes to the central location.

(ii) **Scalability**: WSN implementations must have the feature of scalability associated with their architectures and deployments. In the event of a future increase or decrease of sensor node units, the WSN must support the scaling of the infrastructure without changing the whole implementation.

(iii) **Long lifetime**: The lifetime or the energy renewal cycle of WSNs must be long enough to make large-scale applications feasible.

(iv) **Security**: The security of WSNs, if not considered, can easily harm the security of the whole system, right back to the central server.

Predecessors of IoT/ wireless sensor networks (WSN)

(v) **Programmability**: The programmability of WSNs is important as it ensures the robustness of these systems. WSNs deployed in one application area can be reused for other applications just as easily with a change in sensors and the backend programs associated with it.

(vi) **Affordability** القدرة على تحمل التكاليف : As WSNs generally require multiple units, typically in the range of tens or hundreds of WSN nodes, the cost of the nodes and their affordability is vastly responsible for the acceptability of the system.

(vii) **Heterogeneity** التباين : The WSNs must support a wide number and various types of sensors and solutions, thus enabling heterogeneity. In the absence of heterogeneity, the WSN will tend to become very application-specific, which in turn would require major customizations even in the event of minor changes to the network or architecture.

(viii) **Mobility**: WSNs must support the notion of mobility of nodes such that the nodes may be easily relocatable or mobile. Mobility would ensure the rapid deployability of WSN-based solutions in all environment types.

Predecessors of IoT/ Architectural components of WSN

- The WSN stack is made up of **five layers:**

1) Physical layer.
2) data link layer.
3) Network layer.
4) Transport layer.
5) Application layer.

- In addition to these five layers, the WSN stack further comprises **five cross planes concerned** with **management tasks**:

1) **power** management plane.
2) **mobility** management plane.
3) **task** management plane.
4) **QoS** management plane.
5) **security** management plane.

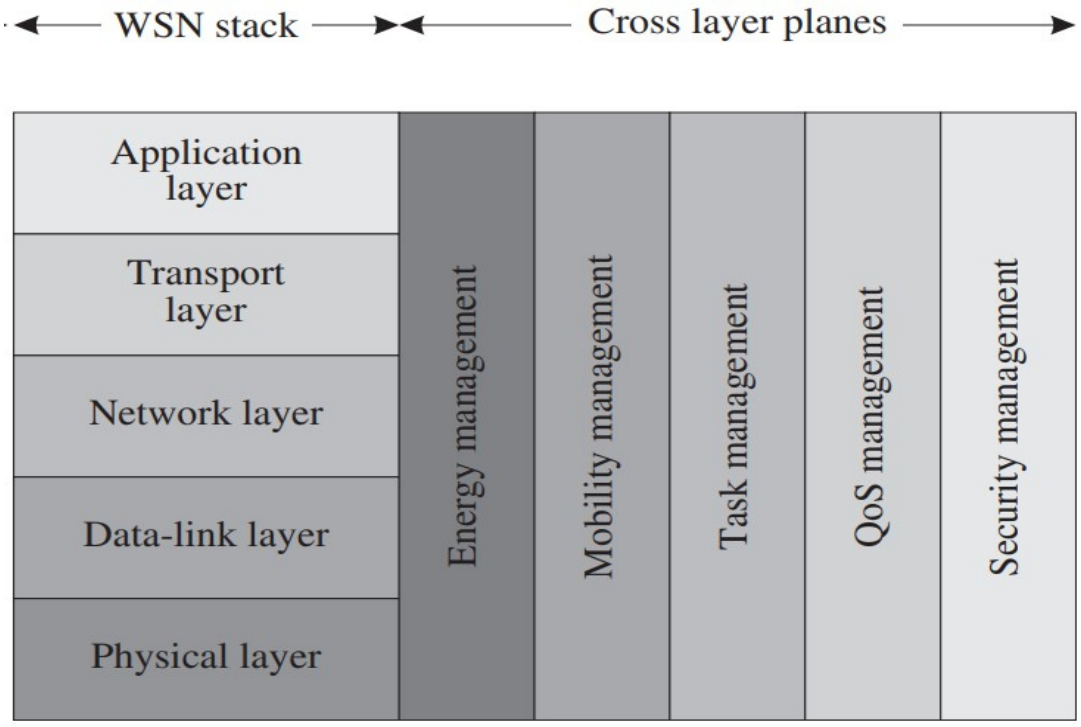


Figure: The various functional layers for a WSN communication and networking architecture

Predecessors of IoT/ Architectural components of WSN

- The **Components of the WSN stack** is made up of **five layers**:

1) Physical layer:

- ❖ is at the **bottom** of the stack and responsible for enabling the transmission of signals over a physical medium between multiple WSN nodes/units.
- ❖ In WSNs, this layer is **responsible** for carrier frequency generation, carrier frequency selection, modulation/ demodulation, encryption/decryption, and signal detection.
- ❖ Typically, WSNs make use of the IEEE 802.15.4 standard for this layer because of its low cost, low energy budget, low data rate, and small form factor.

2) data link layer:

- ❖ resides **above** the physical layer.
- ❖ It is responsible for medium access control (MAC) functions such as multiplexing/ demultiplexing, framing of messages from the upper layer, frame detection, and error control.
- ❖ These **functions** help in ensuring the reliability of communication between the WSN nodes.

Predecessors of IoT/ Architectural components of WSN

- The **Components of the WSN stack** is made up of **five** layers:

3) Network layer:

- ❖ lies on **top** of the **data link** layer.
- ❖ The primary function associated with this layer is the routing of packets.

4) Transport layer:

- ❖ This layer, plays a crucial role in ensuring reliability and congestion control of the packets **arriving** and **leaving** from each WSN node.

5) Application layer:

- ❖ This layer sits on **top** of all the previous **four layers**.
- ❖ Responsible for software interfaces.
- ❖ The software interfaces are responsible for the conversion of data from various application domains of WSN into an acceptable format for transfer to the layer underneath this layer.

Predecessors of IoT/ Architectural components of WSN

- **The five <u>Cross-layer Management</u> Planes**:
  the WSN stack further comprises five cross planes concerned with **management tasks**:

- The use of OSI-like stacks for outlining the functionalities of WSNs faces limitations due to specialized operations of the stacks in areas requiring prolonged deployments with constrained energy and communication infrastructure, and mobility.
- Unlike regular computer networks, the OSI-like WSN stack does not fully describe the functionalities of WSN-based systems.
- This is because its specialized nature results in a strong correlation between the five WSN stack layers.
- Typically, solutions addressing WSN applications and functionalities make joint use of all the five layers.
- It is mainly because of this reason that the cross-layer management plane structure is more popularly accepted as a means of abstraction of WSN-based systems and solutions.

Predecessors of IoT/ Architectural components of WSN

Sl. no.	Features	Energy management	Mobility management	Task management	QoS management	Security management
1	Functionality	Maximizing the energy of WSN nodes and overall network energy management.	Ensuring connectivity even when the WSN nodes are moving.	Distribution of tasks among WSN nodes for ensuring network lifetime.	Ensuring the quality of the service being offered by the WSNs.	Ensuring uncompromising security and integrity of the sensed and transmitted data.
2	Applications	Environment monitoring, Home monitoring, Multimedia sensors	Vehicular monitoring, Unmanned aerial vehicle networks	Environment monitoring, Agricultural monitoring, Underground and underwater sensor networks	Vehicular monitoring, Multimedia sensors	Military sensor networks, Industrial monitoring
3	Tasks	Deciding network size and deployment density.	Topology management, Sensing coverage, Communication coverage	Sensor scheduling, Processing scheduling, Data forwarding	Bandwidth allocation, Resource allocation, Error management	Security of transmission, Node security
4	Challenges	Node deployment density, Sleep scheduling, Maximizing node lifetime	Clustering, Routing	Leader election, Workforce selection	Bandwidth optimization, Jamming avoidance	Low-power security protocols

Predecessors of IoT/ Architectural components of WSN

- **Classes /types of Wireless Sensor Networks :**

A. Some of the applications of WSNs :

i. **Military Applications**: WSNs are used for the **detection of** enemy soldiers, vehicles, intrusion, weapon systems, and armaments.

ii. **Health Applications**: WSNs in healthcare are being used to **monitor patients** in hospitals, ambulances, and homes. Nowadays, a new class of healthcare devices—**wearable appliances**—enable a user to have a miniature health sensor on them without additional discomfort.

iii. **Environmental Applications**: WSNs are used for **environmental monitoring** of pollution, tracking of wildlife, forests, and others.

iv. **Home Applications**: WSNs in the home have given rise to home automation systems and smart home connectivity systems.

v. **Commercial Applications:** WSNs are used for **tracking** vehicles, and packages in transport, logistics, and others.

vi. **Industrial Monitoring:** WSNs in industries **keep track of** various industrial processes, monitor factory floors, ensure worker safety, and perform stock management.

Predecessors of IoT/ Architectural components of WSN

- **Classes /types of Wireless Sensor Networks :**

B. WSNs organized into domains of implementation :

i. Wireless **Multimedia** Sensor Networks (**WMSN**).

ii. U**nderwater** Sensor Networks (**UWSN**).

iii. Wireless **Underground** Sensor Networks (**WUSN**).

iv. Wireless **Mobile** Sensor Networks (**MSN**).

Predecessors of IoT/ Architectural components of WSN

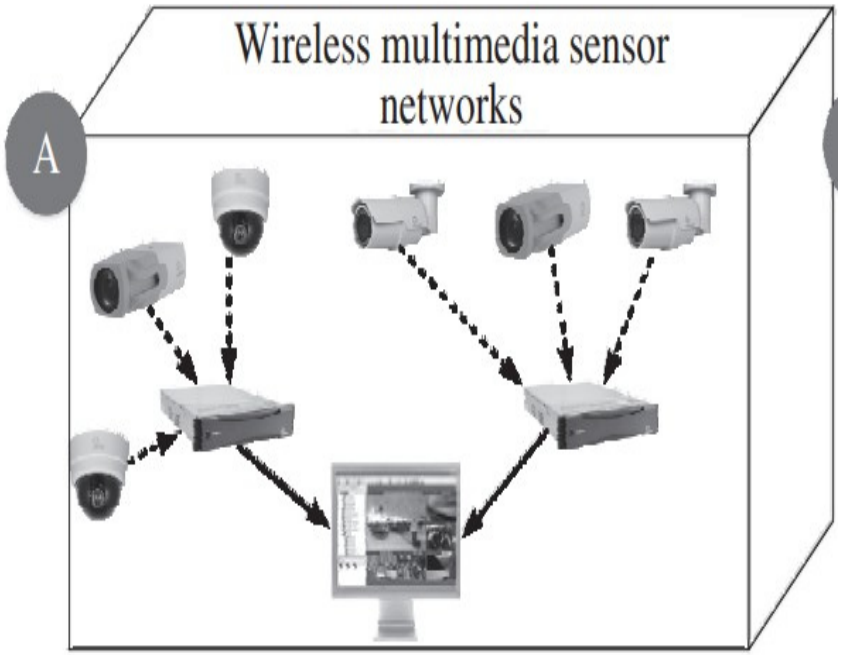
- **Classes /types of Wireless Sensor Networks :**

B. WSNs organized into domains of implementation :

i. Wireless Multimedia Sensor Networks (WMSN):

- This class of WSNs boasts of the ability to retrieve **videos**, **audio**, **images**, or **all three** in addition to regular scalar **sensor readings**.

- However, due to the use of multimedia sensors, the power and processing requirements of this class of WSNs are very high as compared to the other classes of WSNs.



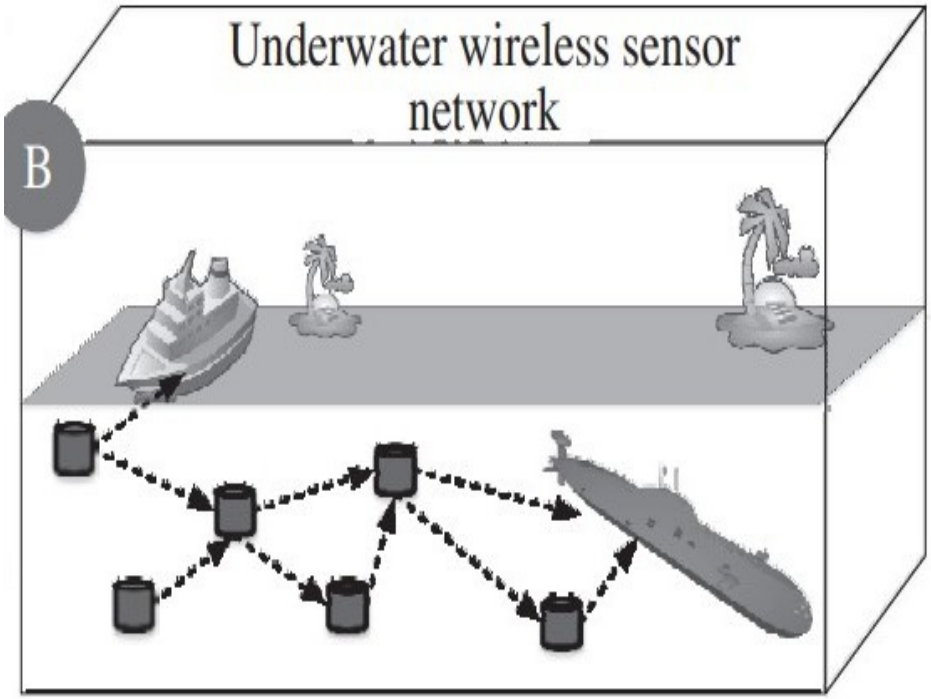
Predecessors of IoT/ Architectural components of WSN

- **Classes /types of Wireless Sensor Networks :**

B. WSNs organized into domains of implementation :

ii. <u>Underwater</u> Sensor Networks (UWSN):

- ❖ This class of WSN is designed specifically to work in underwater environments.
- ❖ However, the long propagation delays and uneven data rate makes it necessary to develop newer topologies and architectures, which can work under the conditions of severe limitations of the physical layer.



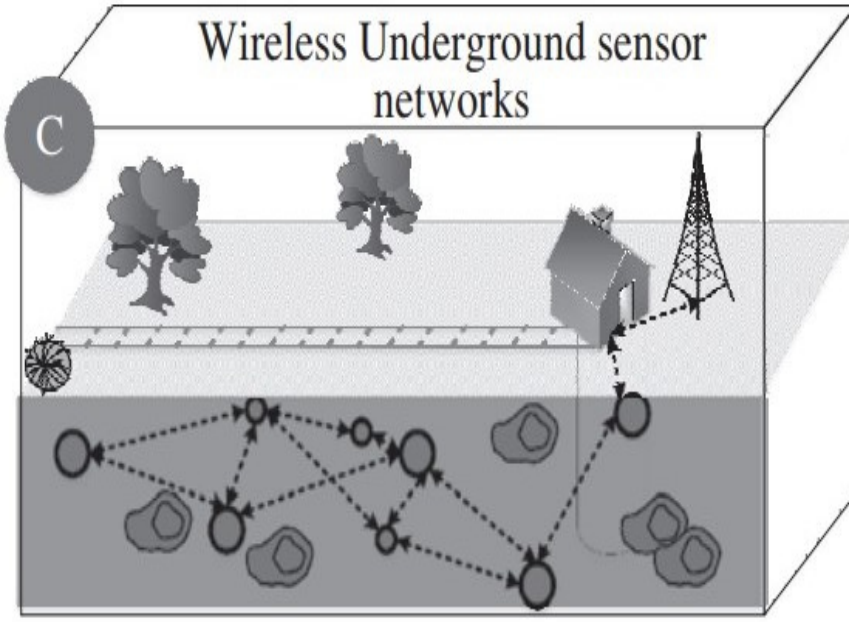
Predecessors of IoT/ Architectural components of WSN

- **Classes /types of Wireless Sensor Networks :**

B. WSNs organized into domains of implementation :

iii. Wireless Underground Sensor Networks (WUSN):

- This class of WSNs is designed to be deployed entirely underground.
- The underground environment poses challenges of attenuation due to the **rocks** and **minerals** in the soil.
- **Another** significant **problem** associated with this class is the need for digging up of the nodes to **replenish their energy sources**.
- Typical usage scenarios of this class of WSNs are underground **mines** and **monitoring** of underground plumbing systems.
- WUSNs need denser deployment architectures owing to the limited range of wireless communication in underground environments.



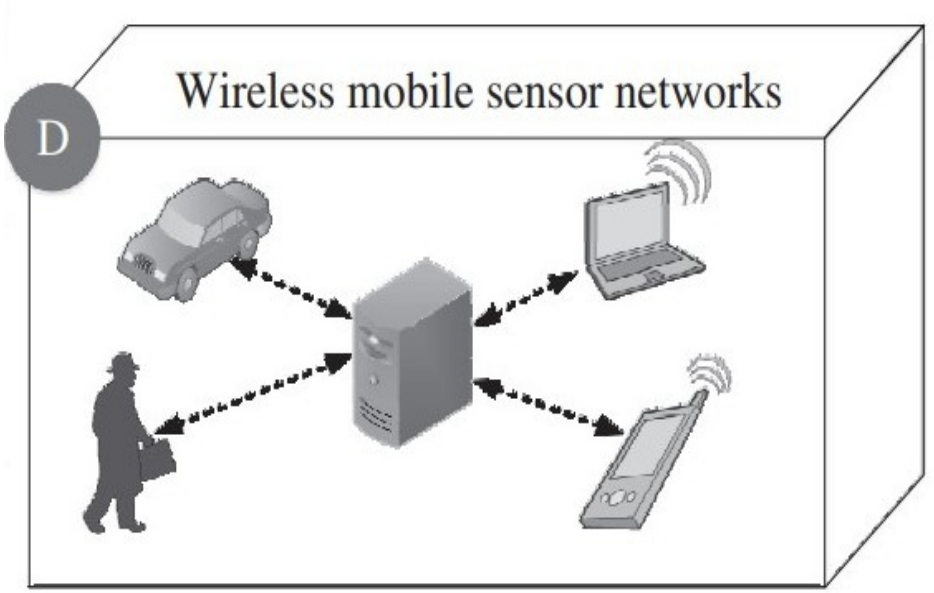
Predecessors of IoT/ Architectural components of WSN

- **Classes /types of Wireless Sensor Networks :**

B. WSNs organized into domains of implementation :

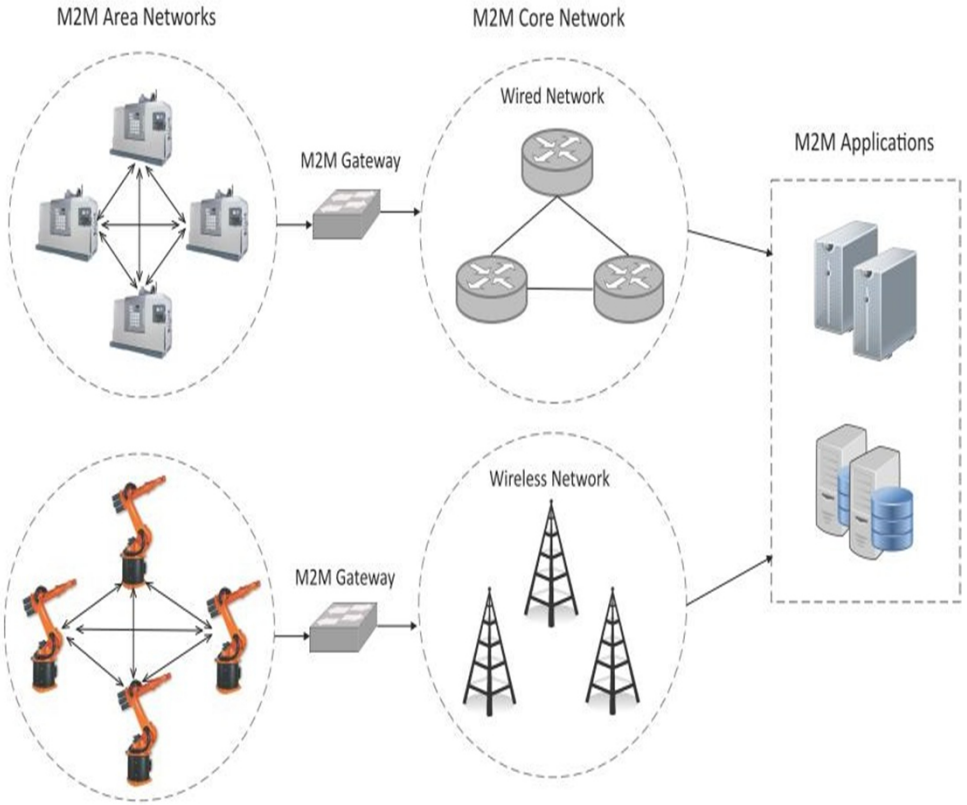
iv. Wireless Mobile Sensor Networks (MSN):

- This class of WSNs is **characterized** by its mobility and low power requirements.
- The **sensor nodes are mobile**, which requires them to rapidly connect to networks, disconnect from them, and then again connect to new networks **until the nodes are mobile**.
- Typical examples of MSNs include smartphone networks, wearables, vehicular networks, and others.



Predecessors of IoT/ Machine-to-Machine Communications

- The machine-to-machine (M2M) paradigm, as the name suggests, implies a system of communication between two or more machines/devices **without human intervention**.
- An M2M area **network** comprises machines (or M2M nodes) that have **embedded hardware modules** for sensing, actuation, and communication.
- Various **communication protocols** can be used for M2M local area networks, these communication Protocols provides **connectivity** between M2M node within M2M area networks.



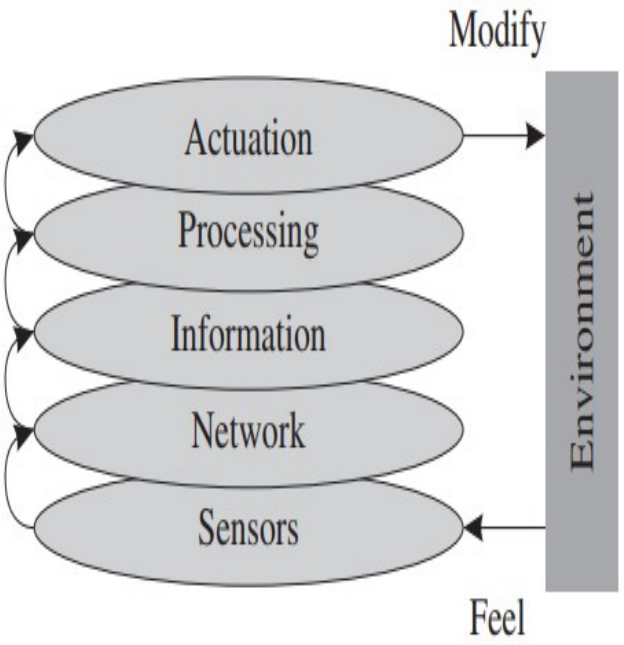
Predecessors of IoT/ Machine-to-Machine Communications

- **Some basic examples** of **M2M communication** in our daily lives include ATM machines signaling banks about the need to refill them with cash, power line monitoring systems in a house alerting a generator set of possible power failures and switching to generator-based supply, vending machines updating stock of items in their inventory and alerting a remote inventory of the need to refill certain depleting items, and others.

- The task of any **sensor network system** (be it WSN, M2M, CPS, or IoT) is to **sense** a physical environment and convert the acquired **data** into a tangible output in the **form of numbers** using sensors.
- This sensing is followed by the **transferring of sense data** to a remote device or location **using a network**, which may be wired or wireless.
- The data collected at the remote device from various sensors—homogeneous or heterogeneous—is converted to usable **information**, which can be utilized to define the course of action for individual scenarios.
- This information is processed to **decide** upon the most valid and optimum course of **action** that must be undertaken to **control the sensed environment** desirably or as per requirements.
- Finally, actuators are put to work to **modify** or **adjust** the **sensed environment**.

Predecessors of IoT/ Machine-to-Machine Communications

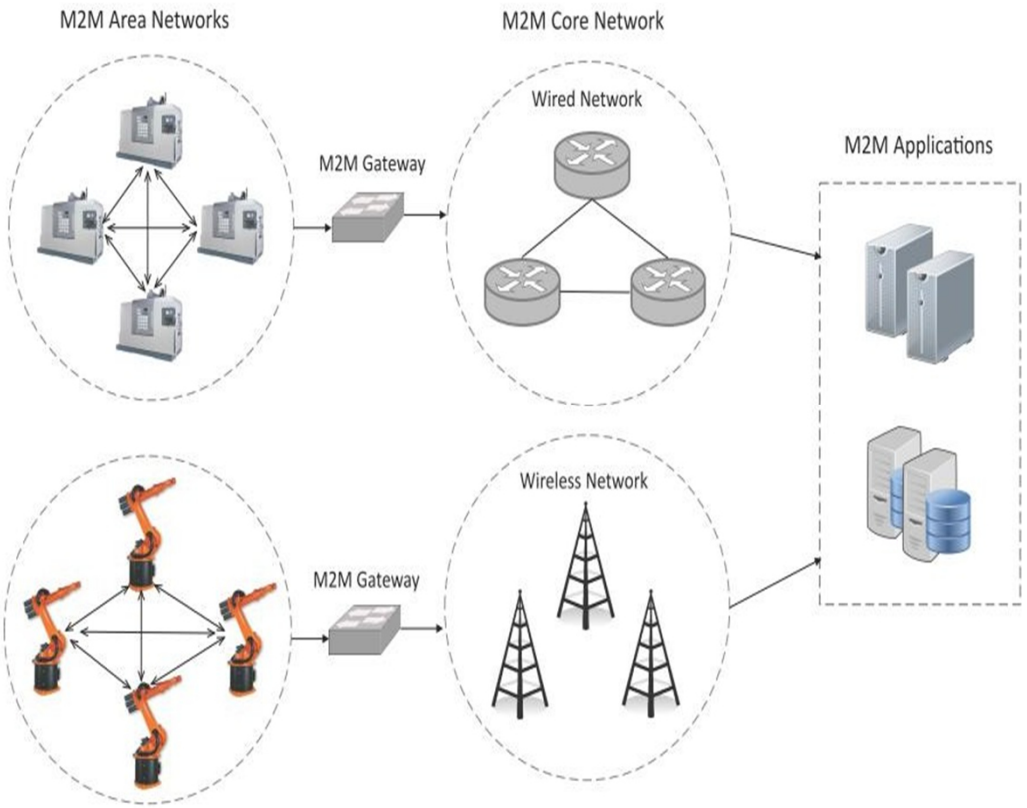
- Figure shows an **overview** of the M2M **ecosystem**, it outlines the **significant aspects** of an M2M ecosystem.
- The task of any **sensor network system** (be it WSN, M2M, CPS, or IoT) is to **sense** a physical environment and convert the acquired **data** into a tangible output in the **form of numbers** using sensors.
- This sensing is followed by the **transferring of sense data** to a **remote device** or location **using a network**, which may be wired or wireless.
- The data collected at the **remote device** from various sensors—homogeneous or heterogeneous—is converted to usable **information**, which can be utilized to define the course of action for individual scenarios.
- This information is **processed** to **decide** upon the most valid and optimum course of **action** that must be undertaken to **control the sensed environment** desirably or as per requirements.
- Finally, actuators are put to work to **modify** or **adjust** the **sensed environment**.



Predecessors of IoT/ Machine-to-Machine Communications

- The communication network in M2M **serves** as a transport medium for exchanging data between two or more devices. It may be wired as well as wireless.

- The massive and rapid developments in the field of wireless communication have significantly helped in the widespread deployment of M2M solutions the world over.



Predecessors of IoT/ Machine-to-Machine Communications

- The standard requirements of an **M2M platform** include **features**:

1) device management.
2) User management.
3) Data management.
4) Web access and Cloud.
5) P2P communication.
6) M2M area network.
7) Connection management.

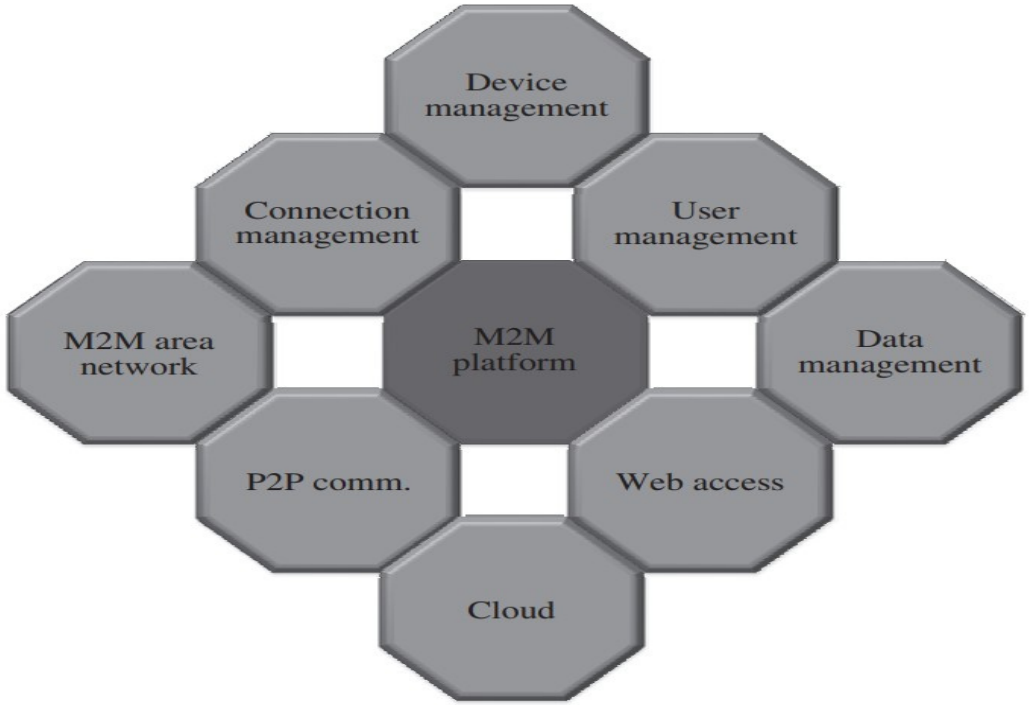


Figure The various features desirable in an ideal M2M platform

Predecessors of IoT/ Machine-to-Machine Communications

- The standard requirements of an **M2M platform** include **features**:

1) **device management: profile** should be available on the platform so that the platform can add, remove, modify, and query **devices** attached to the platform. The **features** of control and authentication of devices also fall under this category.

2) **User management: profile** is **tasked** with user access restrictions, device access permissions, service access permissions, and user authentication and management tasks.

3) **Data management: feature** enables the collection of data from different devices and objects connected to the platform. It should **allow** the user to query the collected data, control the actuators connected to the platform, as well as **allow** the data to dictate the control of connected devices and actuators.

4) **Web access & Cloud: features** are intended to provide anywhere, anytime, and anyhow access to the data and the devices connected to the M2M platform over the Internet.

Predecessors of IoT/ Machine-to-Machine Communications

- The standard requirements of an **M2M platform** include **features**:

5) **peer-to-peer(P2P) communication:** feature **facilitates** the reduction of unnecessary traffic through the M2M platform by reducing unnecessary platform accesses and data storage of the platform.

6) **M2M area network:** management profile facilitates the control of a zonal implementation of M2M, such as in sensor networks.

7) **Connection management:** allows for the interoperability between devices and communication methods by utilizing a single platform.

Predecessors of IoT/ Architectural components of M2M

- M2M being a complex paradigm is **better understood if the components are grouped** under the following **two categories**:

1) **M2M networking model:** The networking model approaches the prospective **scopes and features** of the M2M platform in terms of **the networking components** and their **roles**.

2) **M2M service ecosystem:** The service ecosystem attempts to **describe** the **M2M platform** and **interactions** in terms of the **various service providers**, their **roles** and **responsibilities**.

Predecessors of IoT/ Architectural components of M2M

1) **M2M networking model:** The networking model approaches the prospective **scopes** and **features** of the M2M platform in terms of the **networking components** and their roles.

I. **M2M Devices:**

II. **M2M Area Network.**

III. **M2M Gateway.**

IV. **M2M Communication Network.**

Predecessors of IoT/ Architectural components of M2M

I. M2M Devices:

- **M2M devices** are those entities that are capable of responding to requests for data by means of replying through networked messages almost independently.
- The devices at the end of the network, which are tasked with sensing and actuation, also fall under this category.
- The mode of communication of these devices may be wired or wireless.
- They connect to a network through a gateway, or directly using a cellular operator's network.

Predecessors of IoT/ Architectural components of M2M

- M2M **devices** can be broadly **categorized** into **three types** based on their interaction and features concerning an M2M platform.

1. **Low-end Devices.**
2. **Mid-end Devices.**
3. **High-end devices.**

Predecessors of IoT/ Architectural components of M2M

1. **Low-end Devices:** This device type is typically **cheap** and has **low capabilities** such as auto-configuration, power saving, and data aggregation. As devices of this type are generally static, energy-efficient, and simple, a highly dense deployment is needed to increase network lifetime and survivability. Moreover, low-end devices are resource-constrained with no IP (Internet protocol) support; they are generally used for environmental monitoring applications.

2. **Mid-end Devices:** These devices are **more costly than low-end M2M** devices as they may have mobility associated with them. However, these devices are less complex and energy-efficient than high-end devices. The presence of capabilities such as localization, intelligence, support for quality of service (QoS), traffic control, TCP/IP support, and power control makes them appealing for applications such as home networks, SCM (supply chain management), asset management, and industrial automation.

3. **High-end devices:** These devices generally require low density of deployment. They are designed such that they **can handle multimedia data** (videos) with QoS requirements, even in mobile environments. The mandatory inclusion of mobility as a feature of these devices makes them **costly**. Generally, they are **used for** ITS (intelligent transportation system) and military or bio-medical applications.

Predecessors of IoT/ Architectural components of M2M

II. M2M Area Network:

- The M2M area network comprises multiple M2M devices, **either** communicating with one another **or** to a connected platform. The local **communication** between the **M2M devices** up to the **M2M gateway** can be considered as the M2M area network. It is also referred to as **the device domain**. Some examples that can be correlated to the functioning of M2M area networks include personal area networks (PANs) and local nodes in a wireless sensor network (WSN).

III. M2M Gateway:

- It is responsible for **enabling connectivity and communication** between the M2M **devices** and a **global** communication channel such as the **Internet**. The **gateway** is responsible for distinguishing between data and control signals on the **M2M platform** to enable monitoring as well as maintenance of the M2M area network remotely. The **gateways** must additionally ensure that the M2M devices can access an outside network and that the **devices** themselves can be accessed from an outside network.

Predecessors of IoT/ Architectural components of M2M

IV. M2M Communication Network:

❖ This is also referred to as the **M2M network domain**. It consists of the communication technologies and paradigms for enabling connectivity and communication between M2M gateways and various applications. These **M2M networks** can be **classified** as either

1. **IP**-based or
2. **non-IP**-based.

Predecessors of IoT/ Architectural components of M2M

IV. M2M Communication Network:

1. IP-based Networks:

- ✓ These **networks** are supported only by high-end M2M devices. As the other two M2M device types—**low-end** and **mid-end**—are typically resource-constrained, both **IPv4** and **IPv6** are supported in IP-based M2M networks.

- ✓ However, **IPv6**-based schemes are **preferred** due to the provision of **scalability** in IPv6-based communication, which is **absent** in IPv4.

- ✓ The **figure** shows communication of IP-based M2M networks. As both the **M2M network** and the **IP network** (which is global/ accessible by the Internet) follow **similar stack structures**, the communication between them is direct, without the need for adaptors or protocol tunnels.

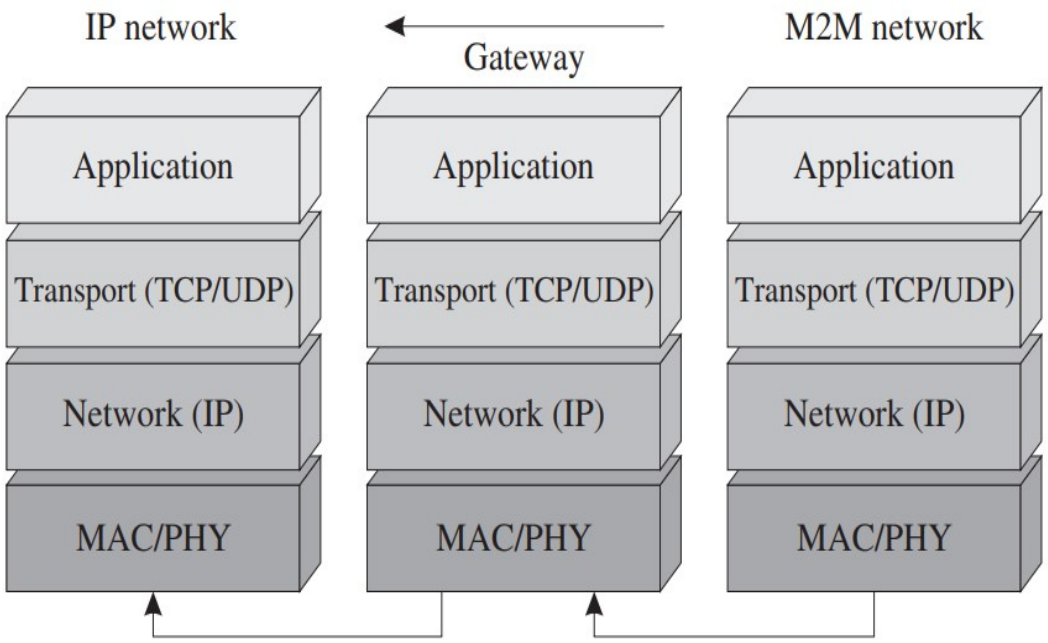


Figure M2M communication over an IP-based network

Predecessors of IoT/ Architectural components of M2M

IV. M2M Communication Network:

2. Non-IP-based Networks:

- ✓ This scheme is generally used with low-end and mid-end M2M devices.
- ✓ A **separate addressing scheme** is designed for accommodating وفق بين these resource-constrained devices and is limited to the domain within the M2M gateways.
- ✓ The **figure** shows the **non-IP**-based network communication.
- ✓ **The packets** from the resource-constrained M2M devices within the **M2M area network** forward their packets to the **gateway**, which again packetizes them into IP-based packets for further transmission to an IP network.

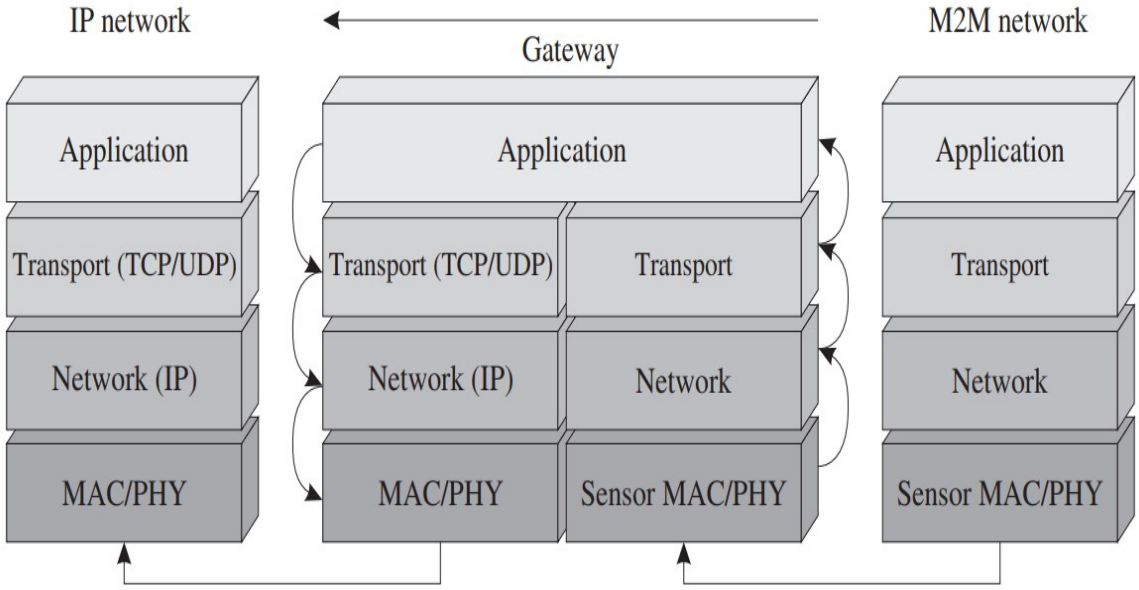


Figure M2M communication over a non-IP-based network

Predecessors of IoT/ Architectural components of M2M

2) M2M service ecosystem:

- The service ecosystem attempts to **describe the M2M platform** and interactions in terms of the **various service providers**, their roles and responsibilities.
- The M2M service ecosystem, unlike the networking model, **classifies** the various **components** of the system based on the needs of the service **offerings** from the M2M platform.
- **The ecosystem** can be broadly divided into **four domains**:

1) M2M area networks.
2) Core network.
3) M2M service platform.
4) Stakeholders.

Predecessors of IoT/ Architectural components of M2M

2) M2M service ecosystem:

❖ The ecosystem can be broadly divided into **four domains**. The functions, and roles of the various domains are as follows.

1) M2M area networks.
2) Core network.
3) M2M service platform.
4) Stakeholders.

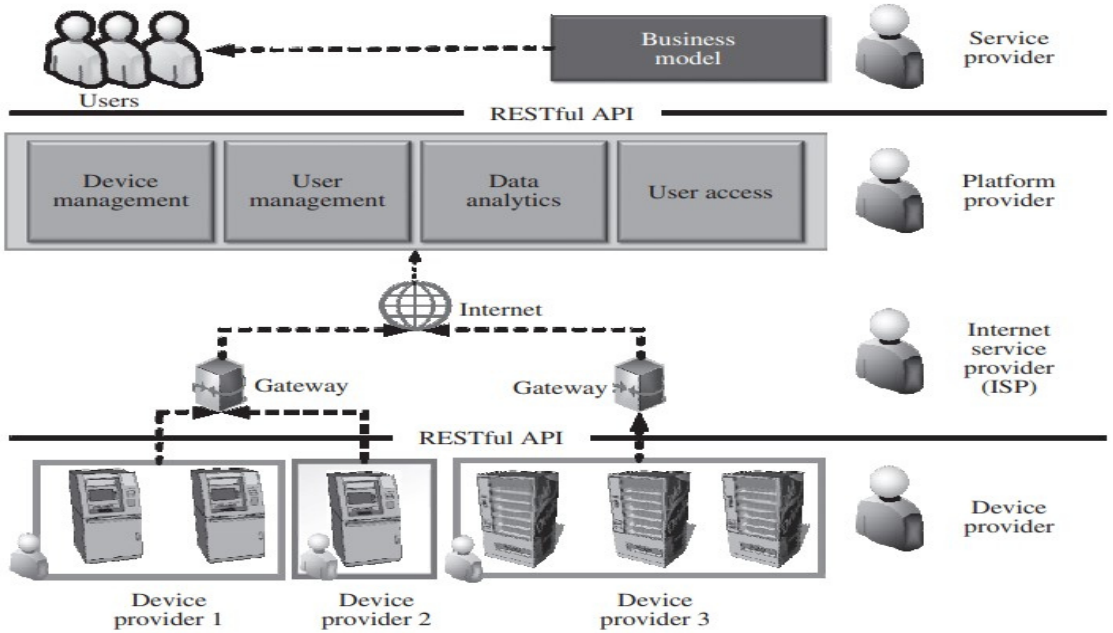


Figure The M2M service platform components and ecosystem

Predecessors of IoT/ Architectural components of M2M

1) M2M area networks.

- These networks form the base of the M2M ecosystem.
- They are **similar** to the M2M area networks previously described in the **M2M networking model**.
- The constituent **devices** are classified as low-end, mid-end, and high-end based on their **functionalities** and ability to handle mobility.

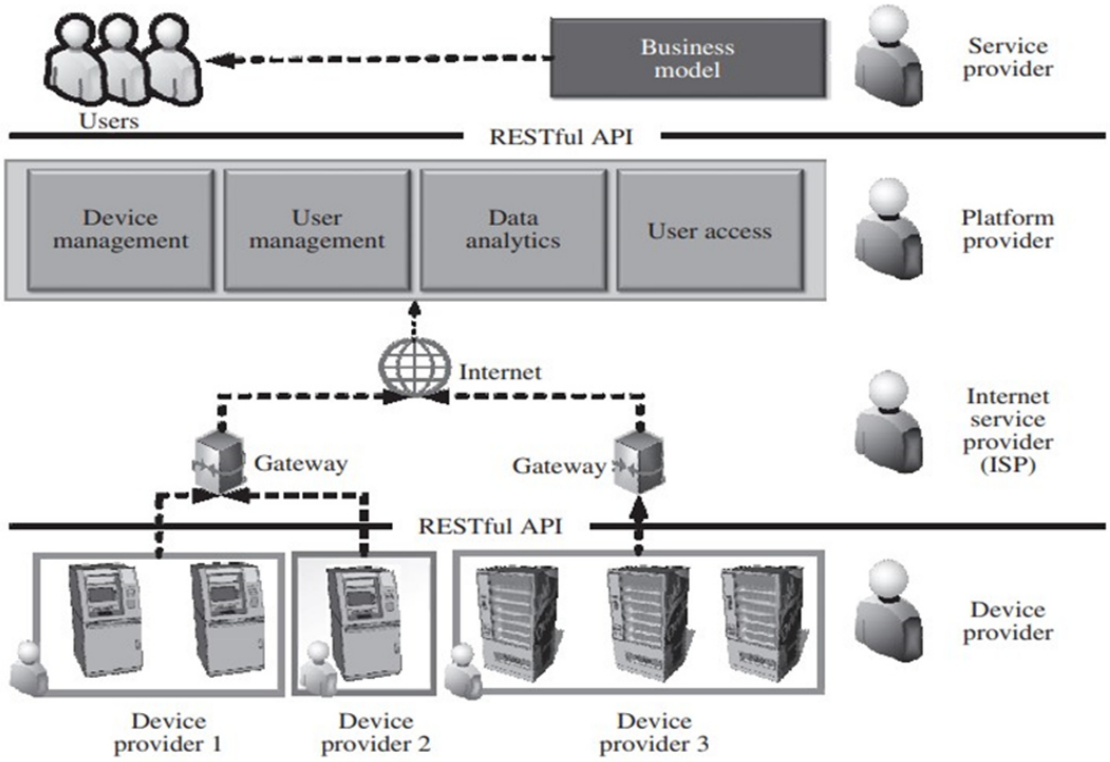


Figure The M2M service platform components and ecosystem

Predecessors of IoT/ Architectural components of M2M

1) M2M area networks.

2) Core network:

- The **core networks** form the basis of the communication infrastructure of the M2M ecosystem, and carry the **bulk** الجزء الأكبر **load of traffic** across the M2M network.

- The **core network** can be **wired** or **wireless** or **both**.

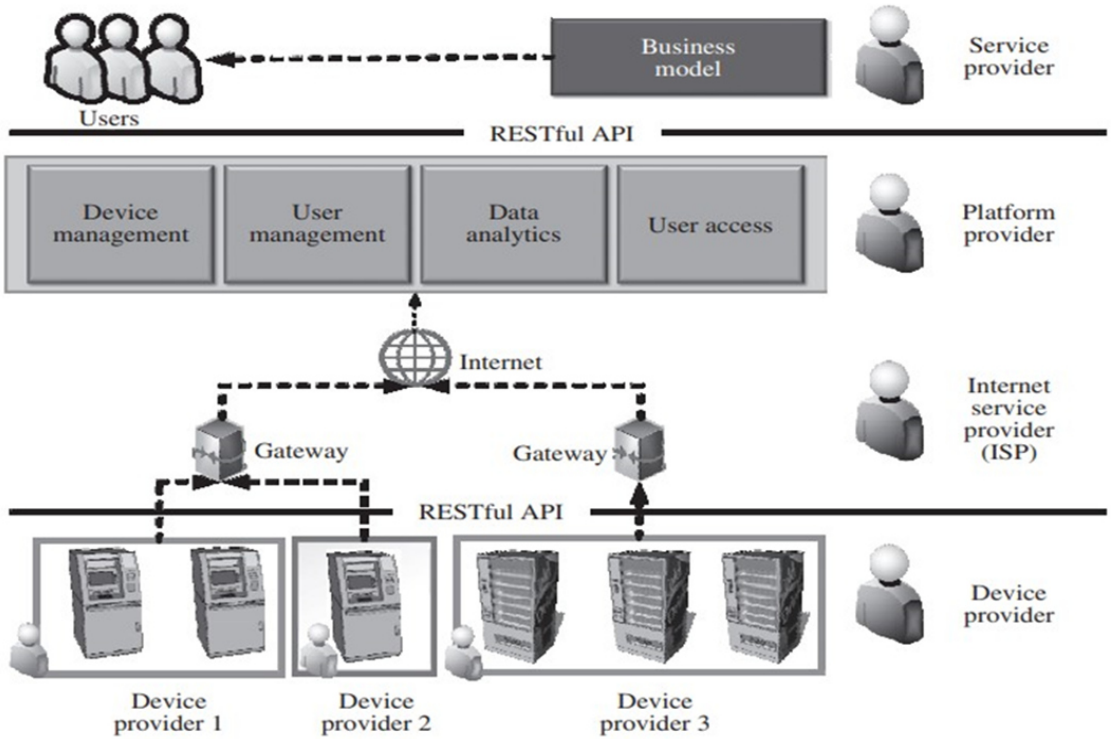


Figure The M2M service platform components and ecosystem

Predecessors of IoT/ Architectural components of M2M

1) M2M area networks.
2) Core network.
3) **M2M service platform:**

❖ The M2M service platform is **further divided** into the following **four** parts:

I. Device Management Platform:

- ✓ This platform enables anytime anywhere access between Internet-connected platforms and registered objects or devices connected to the platform.
- ✓ During device registration, an object database is created from which information can be easily accessed by end users such as managers, users, and services.
- ✓ **The main functions** of this platform include the management of device profiles (location, device type, address, and description), authentication, authorization, and key management functionalities.
- ✓ **Additionally**, it monitors device statuses, M2M area networks, and their interactions and controls.

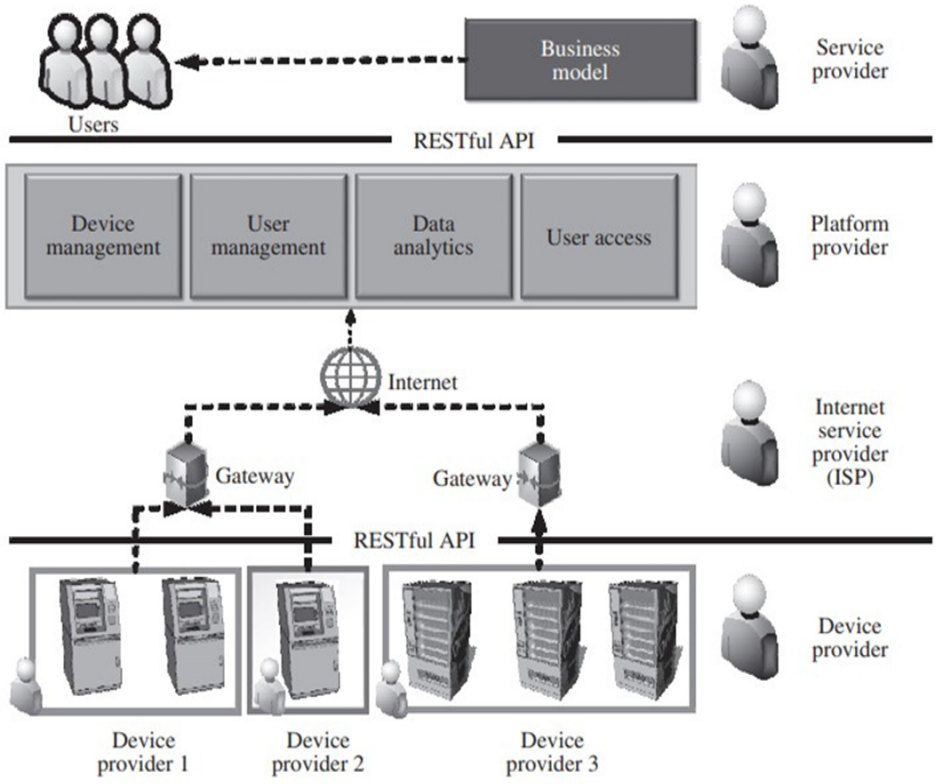


Figure The M2M service platform components and ecosystem

Predecessors of IoT/ Architectural components of M2M

1) M2M area networks.
2) Core network.
3) **M2M service platform:**

II. User Management Platform:

- ✓ Various service providers and device managers can maintain administrative privileges over the devices or networks under their jurisdiction through the platform's device monitoring and control.
- ✓ User profiles and functionalities such as user registration, account modification, service charging, service inquiry, and other M2M services are provisioned and managed through this entity.
- ✓ The platform also enables interoperability between device managers; it provisions control services such as user access restrictions to devices, networks, or/and services.

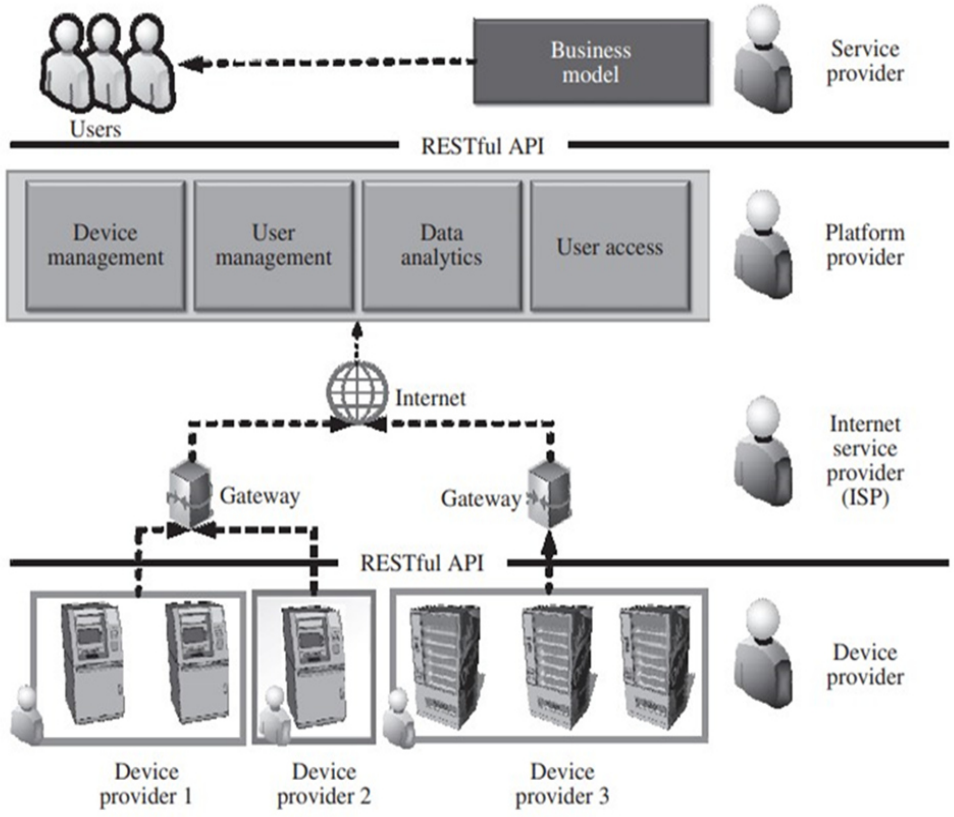


Figure The M2M service platform components and ecosystem

Predecessors of IoT/ Architectural components of M2M

1) M2M area networks.
2) Core network.
3) **M2M service platform:**

III. Data and Analytics Platform:

- ✓ This platform provides integrated services based on device-collected data and datasets.
- ✓ Heterogeneous data emerging from various devices are used for creating new services.
- ✓ This platform collects and controls processing and log data for management purposes.
- ✓ These collected data on the devices are achieved in conjunction with the device management platform.
- ✓ Connection management services, by means of **connecting with the appropriate network**, provide seamless services; this is achieved by analyzing the log and data behavior of the registered devices and networks.

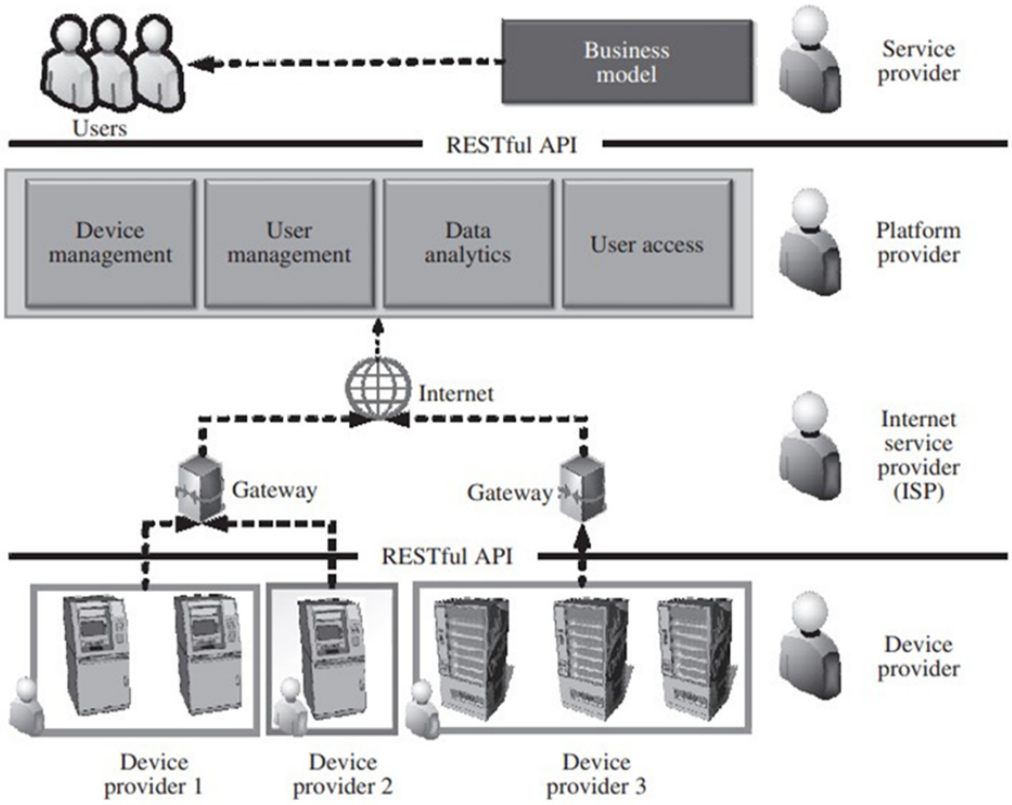


Figure The M2M service platform components and ecosystem

Predecessors of IoT/ Architectural components of M2M

1) M2M area networks.
2) Core network.
3) **M2M service platform:**

IV. User Access Platform:

- ✓ This platform provides a smartphone and web access environment to users.
- ✓ It redirects requests to **service providers** who have a **mapping of the registered devices**, users, and the services subscribed.
- ✓ Provisions for modifications to a device or user-specific mapping are also provided.

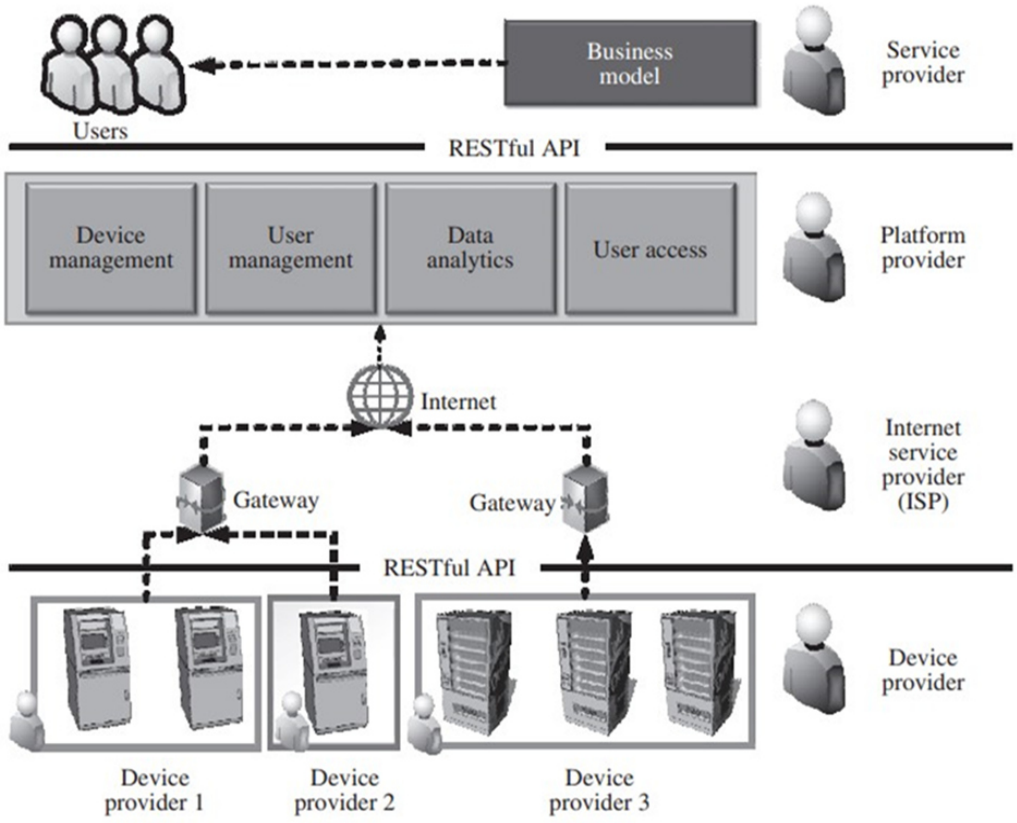


Figure The M2M service platform components and ecosystem

Predecessors of IoT/ Architectural components of M2M

1) M2M area networks.
2) Core network.
3) M2M service platform.
4) **Stakeholders**:

❖ **The stakeholders** in an M2M service ecosystem can be divided into **five different types**:

- ✓ Device providers, Internet service providers (ISPs), Platform providers, Service providers, and Service users.
- ✓ The functional jurisdiction of each of these five classes of stakeholders is well-defined and devised in such a manner that they do not overlap and may be considered mutually exclusive in terms of their offerings.
- ✓ However, at the time of functioning, all these stakeholders have to work together to ensure the smooth functioning of the M2M service ecosystem.
- ✓ Each of these stakeholders and their domains is outlined in **Figure**.

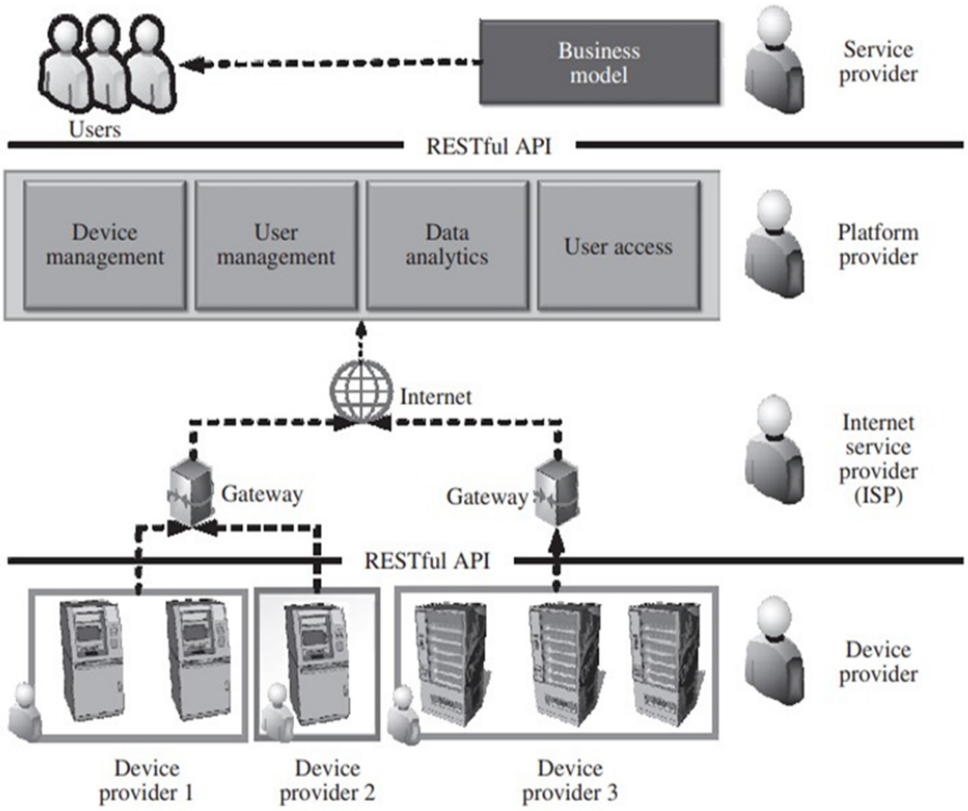


Figure The M2M service platform components and ecosystem

Predecessors of IoT/ Differences between M2M and IoT

- Though both M2M and IoT involve the networking of machines or devices, they differ in the **underlying technologies**, **system architectures**, and **types of applications**.

- Communication Protocols.
- Machines in M2M vs Things in IoT.
- Hardware vs Software Emphasis.
- Data Collection & Analysis.
- Applications.

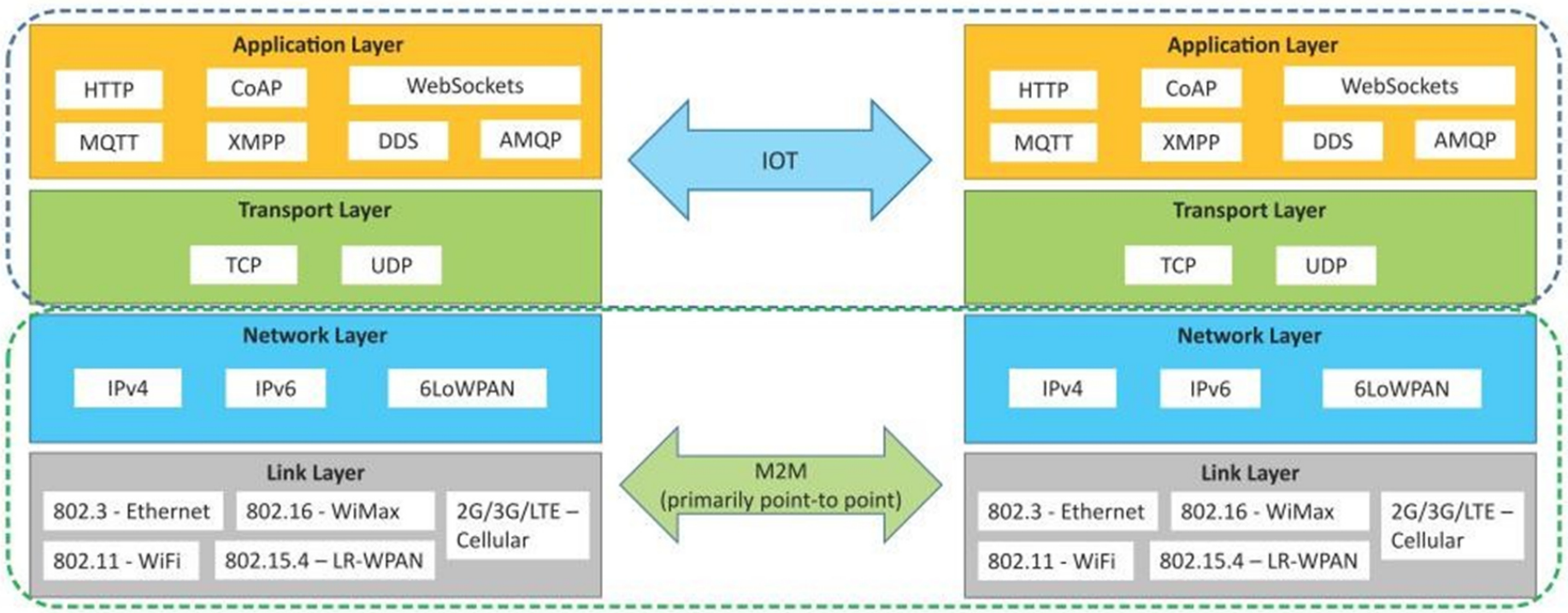
Predecessors of IoT/ Differences between M2M and IoT

❑ Communication Protocols

- M2M and IoT can differ in how the communication between the machines or devices happens.
- **M2M** uses either **proprietary** or **non-IP-based** communication protocols for communication within the M2M area networks.
- The **focus** of **communication** in **M2M** is usually on the protocols **below** the network layer (Physical Network, data link).
- The **focus** of **communication** in **IoT** is usually on the protocols **above** the network layer (Transport, Application).

Predecessors of IoT/ Differences between M2M and IoT

- **Communication Protocols**



Predecessors of IoT/ Differences between M2M and IoT

❑ Machines in M2M vs Things in IoT

- The "**Things**" in IoT refer **to physical objects** that have **unique identifiers** and can **sense** and **communicate** with their **external** environment (and **user applications**) or their internal physical states.
- The unique identifiers for the **thing in IoT** are the **IP addresses** or **MAC addresses**. Things have **software components** for accessing, processing, and storing sensor information, or controlling actuators connected.
- **IoT systems** can have **heterogeneous** things (e.g., home automation IoT systems can include IoT devices of various types, such as fire alarms, door alarms, lighting control devices, etc.)
- **M2M systems**, in contrast to IoT, typically have **homogeneous** machine types within an M2M area network.

Predecessors of IoT/ Differences between M2M and IoT

❑ Hardware vs Software Emphasis

- While the emphasis تشديد ,توكيد of **M2M** is more on **hardware** with **embedded modules**, the emphasis of **IoT** is more on **software**.
- **IoT devices** run specialized **software** for sensor **data collection**, **data analysis** and **interfacing** with the **cloud** through **IP-based** communication.

❑ Data Collection & Analysis

- **M2M** data is **collected** in point solutions and often in **on-premises** storage infrastructure. In contrast to M2M, the data in IoT is **collected** in the **cloud** (can be public, private, or hybrid cloud).
- The **analytics** component analyzes the data and stores the results in the cloud database.
- The IoT data and analysis results are visualized with cloud-based applications.
- The centralized controller is aware of the status of all the end nodes and sends control commands to the nodes.
- Observer nodes can process information and use it for various applications, however, observer nodes do not perform any control functions.

Predecessors of IoT/ Differences between M2M and IoT

❑ Applications

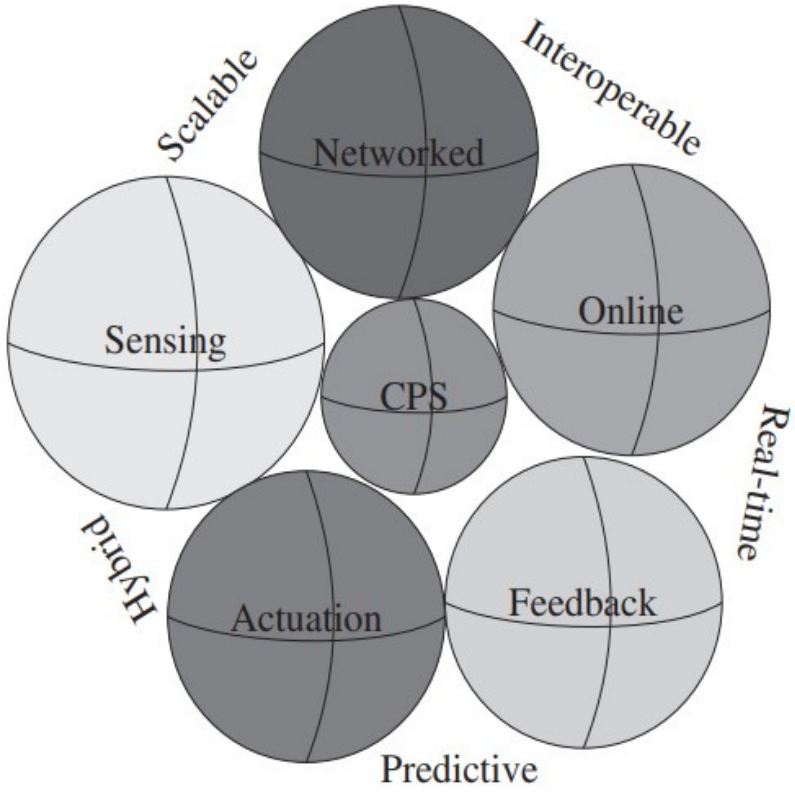
- **M2M** data is collected in point solutions and can be accessed by **on-premises applications** such as diagnosis applications, service management applications, and on-premises enterprise applications.
- **IoT** data is collected in the cloud and can be accessed by **cloud applications** such as analytics applications, enterprise applications, remote diagnosis and management applications, etc.
- Since the scale of **data collected** in **IoT** is so **massive**, cloud-based real-time and batch data analysis frameworks are used for data analysis.

Predecessors of IoT/ Cyber Physical Systems

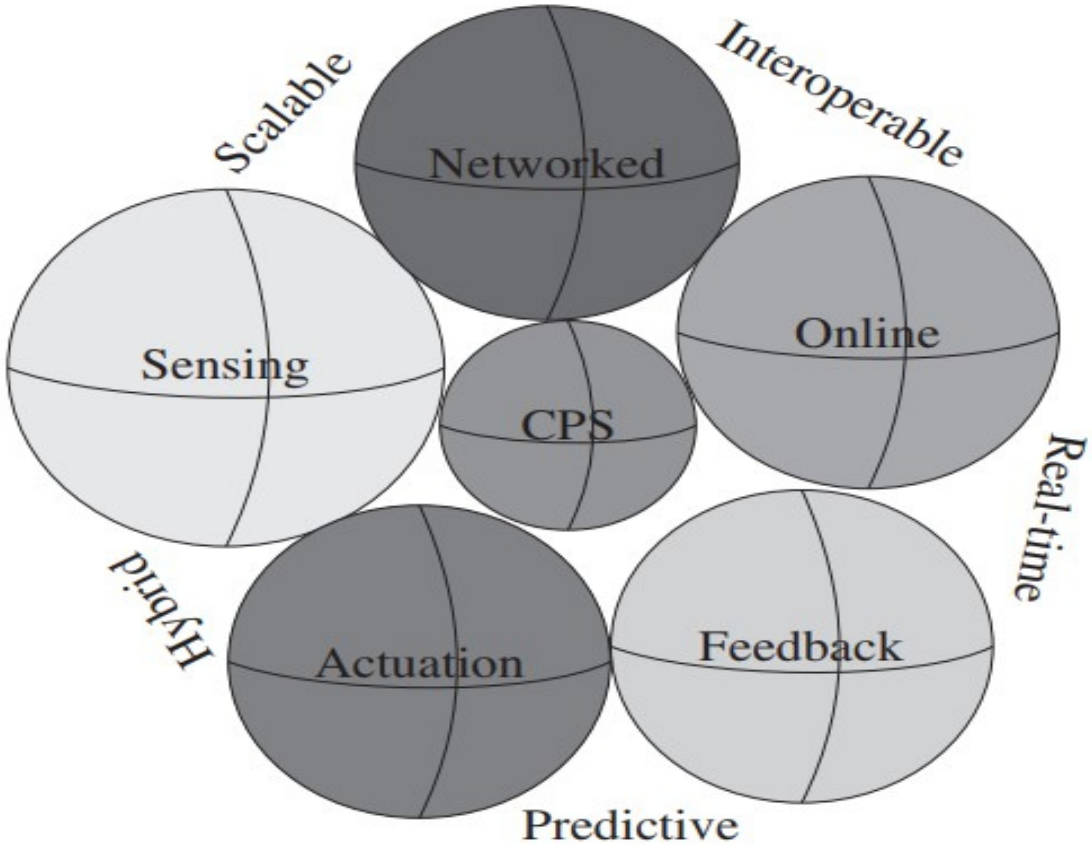
- **Cyber physical systems (CPS)** are Internet-based and networked **monitoring** and **controlling systems** regulated and governed by feedback-based intelligent control algorithms.
- These systems **work** in a highly interdisciplinary domain that involves expertise in lots of domains such as mechanical, electrical, computing, electronics, and many more.
- **They are mainly designed to** **monitor** and **control** physical world processes linked to businesses and industries.
- The most interesting aspect of CPS is the involvement of the concept of human-in-the-loop, which is an integral part of many CPS-based solutions.
- The human-in-the-loop concept simply signifies the involvement of humans in the CPS control cycle. The striking difference of CPS from paradigms such as WSN and M2M is the inclusion of a compulsory feedback system.

Predecessors of IoT/ Cyber Physical Systems

- The typical functioning الاداء النموذجي of a CPS includes the **components** shown in Figure.
- **The sensing mechanism** senses an environment. Various **networked** sensors at the same time generate data for the environment, which is sent over **the Internet** to a processing controlling unit.
- Depending on the intelligent monitoring and control algorithms in the control unit, **feedback** is provided to the **actuators** controlling the state of the environment for which the sensed data was transmitted.
- The changes are again sensed and forwarded to the controller via the previously defined flow.
- The algorithms decide whether the desired state of the environment is achieved or not; they keep sending adjusted feedback to the actuators until the desired state is achieved.



Cyber Physical Systems / The basic overview of CPS features



Predecessors of IoT/ Cyber Physical Systems

- **CPS** is used in a vast range of applications such as backhaul communications, smart grids, healthcare, industrial manufacturing, smart homes and buildings, military and surveillance, robotics, and even transportation. The following **features** generally characterize CPS.

1) Real timeliness.
2) Intelligence.
3) Predictive.
4) Interoperable.
5) Heterogeneous.
6) Scalable.
7) Secure.

Predecessors of IoT/ Cyber Physical Systems

1) **Real timeliness**: CPS depends on real-time communication, processing, and feedback to **effectively provide control to the environment they are deployed in**. For example, in a CPS-based industrial chemical concentration monitoring system, the real-timeliness of the process from sensing to feedback to actuation is crucial for maintaining the operations of the chemical manufacturing plant and preventing disasters.
2) **Intelligence**: Intelligent and adaptive decision making is crucial for the maintenance of CPS-based functionalities. If random or sudden changes in the environment need to be controlled, this feature ensures effective control of the environment and effective coordination between the various dependent subprocesses and systems. For example, in case of an electrical fault in a section of a smart-grid system, the intelligence feature would enable the re-routing of the electrical supply flow through other paths instead of bringing the whole system to a complete standstill.
3) **Predictive**: This feature enables the prediction of outputs and events based on past behavior under similar constraints and conditions. The prediction of events enables the activation of precautionary measures to control the damage if the harmful event does occur. For example, the trend of minor line disturbances and noise in a communication channel might lead to network data loss in a backhaul network. The predictive feature would help in the timely activation of preventive countermeasures to avoid network outage in case the network does start massively dropping packets.

Predecessors of IoT/ Cyber Physical Systems

4) **Interoperable**: The vast and massive deployment zones of CPS-based systems may include software as well as hardware from a variety of manufacturers. This would lead to data, speed, and format **mismatch** under normal circumstances; however, CPS-based **interoperability** prevents this and enables systems from various vendors to **work in sync with one another** as a single system. **Interoperability** also ensures that legacy systems already in place are **not replaced but are added** to the CPS infrastructure.

5) **Heterogeneous**: Heterogeneity in CPS-based systems may be in the types of **actuators**, **sensors**, **processors**, and **data formats being used**, besides the sensing types, software, and application types in a given CPS deployment. However, provisions are already present in CPS to accommodate these types of challenges.

6) **Scalable**: Scalability in CPS-based systems may be in terms of network bandwidth being required due to various sensing types (scalar or multimedia), number of sensors and actuators, size of deployment zones, and other factors. **CPS systems** should be able to handle such demands even after preliminary deployment. **For example**, a smart building wants to incorporate human presence detectors to control the central cooling for the whole building. Initially, conventional scalar sensors were deployed on all floors and corridors to monitor the approximate headcount of people in the building.

Predecessors of IoT/ Cyber Physical Systems

However, after some years, the building management upgrades the scalar sensors by replacing them with camera sensors. Cameras generate huge volumes of data as compared to the scalar sensors previously used. The deployed CPS should be able to accommodate this upgrade **without changing the whole system**.

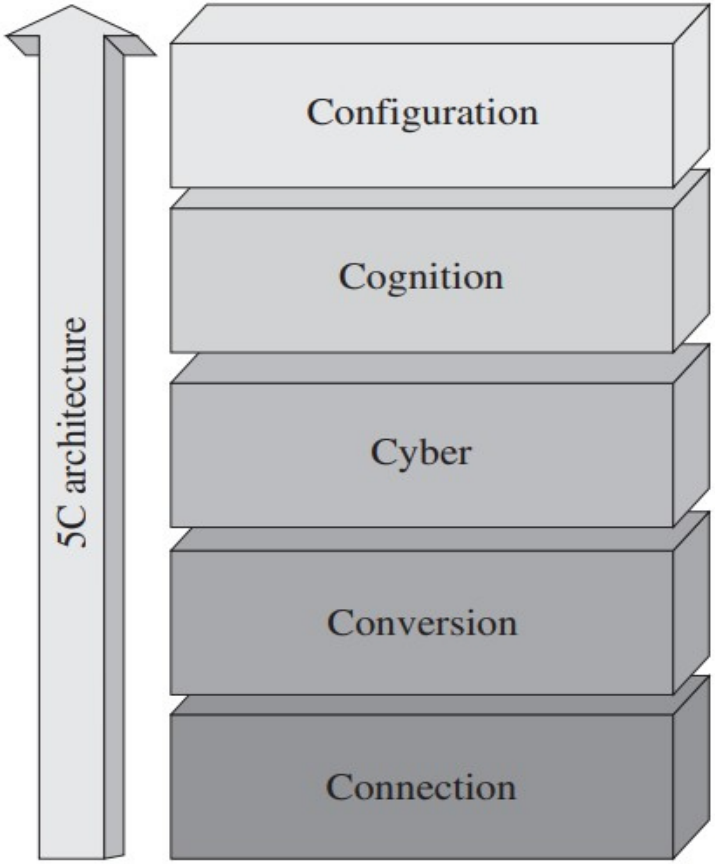
7) **Secure**: The security of CPS is crucial as almost all of the traffic flows through a network and eventually over the Internet. Provisions should be in place to avoid unauthenticated use of the CPS and its hijacking by unscrupulous elements عديمة الضمير عناصر قبل من اختطافه or even attacks, which may reduce the response of the system or eventually bring it down all together.

Cyber Physical Systems / Architectural Components of CPS

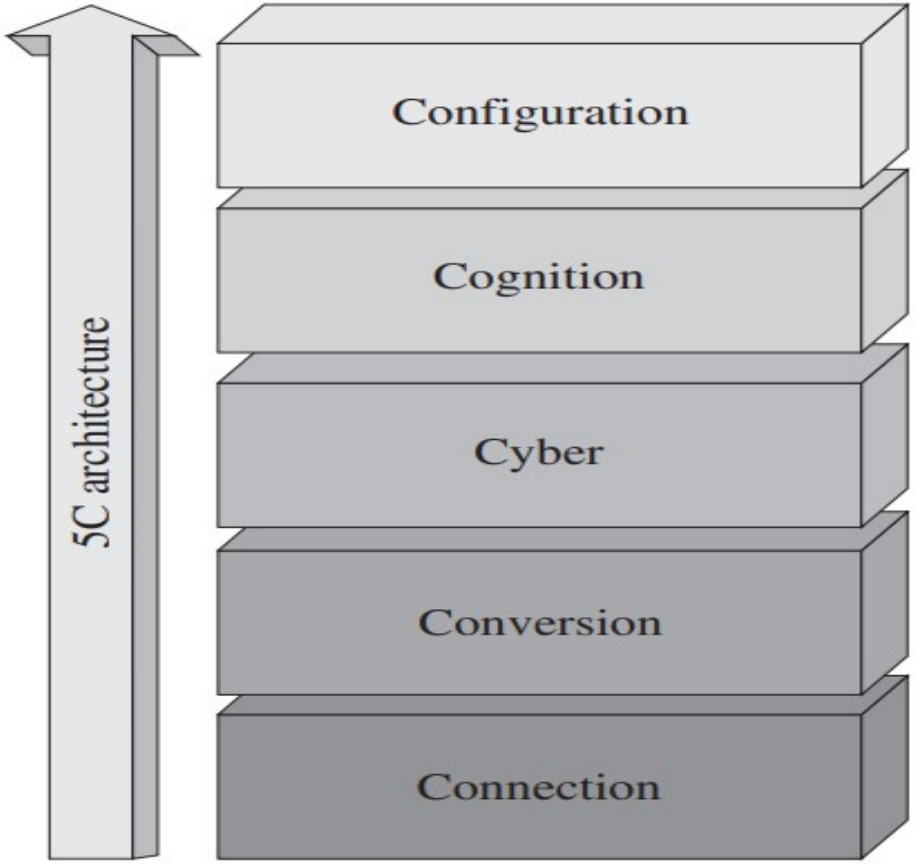
- One of the most accepted CPS architectures is termed the **5C** architecture.

1) **Connection.**
2) **Conversion.** تحويل
3) **Cyber.**
4) **Cognition.** معرفة ادراك
5) **Configuration.**

- Figure describes the CPS **control flow** and **functionalities**.



Architectural Components of CPS / The 5C architecture for CPS



Cyber Physical Systems / Architectural Components of CPS

- Each of these Cs can be described as follows:

1) **Connection**: The **sensed data** from the **base of the architecture** should be accurate and reliable enough to actuate effective feedback for the whole system. The **sensed data** from various **sensor units** should be collected in a hassle-free خالية من المتاعب and organized. The **best possible solution** is the use of tether-free الخالية من القيود communication systems, which should be able to support the plug-and-play التوصيل والتشغيل features of these sensing units.

2) **Conversion**: The **collected data** should be converted to a standard unified format. Post data standardization, **usable information** must be extracted from the sensed data. **Data** from various sensor types and sources need to be correlated تربط to generate practical information from vastly multi-dimensional data. This data can be used to predict للتنبؤ changes to the monitored environments, machinery malfunctions أعطال الآلات , and failures.

Cyber Physical Systems / Architectural Components of CPS

3) **Cyber:** This acts as the **central nodal point** of data collection and the holistic analysis of the system under the control of the CPS. **Data** from various **machine networks**, **environments**, **systems**, and **processes** arrive at this point. Detailed and advanced analytics on the obtained data is performed to gather **statistical trends**. **These trends** can be used to predict the future behavior of machine systems and processes. **The prediction** can be based on digital twins of the actual systems, comparative performance of a machine with other machines, and temporal and regression results of machine health and performance.

4) **Cognition:** This level is mainly **responsible for** the integration of the collective health of the running systems and processes. **The information is presented in the form of human-readable** visualizations and trends. This helps in prioritizing actions and control of processes and systems under the purview of the CPS.

5) **Configuration:** This stage is **responsible for** generating feedback for adjusting the environment being controlled. **The feedback systems** need to be highly adaptive, self-configuring, and resilient for effective control of the system as a whole.

Acknowledgment

- **These lecture slides are based on:**

1) **Chapter 3 (P 48-71)** from the book “Introduction to IoT” by (Sudip Misra, Anandarup Mukherjee, Arijit Roy).
2) **Chapter 3 (P 77-80)** Internet of Things A Hands-On Approach by Arshdeep Bahga, Vijay Madisetti

Basics of Networking

END OF LECTURE (3)

Keep connected with the classroom

btukscx

THANK YOU FOR YOUR ATTENTION