UNIT I

Introduction to Internet of Things

1. INTRODUCTION: The internet that all of us we use is basically a global network of connected standard devices like computers, mobiles & tablets that is governed by standard protocols. Through the Internet, people can share information and communicate from anywhere with an Internet connection.

Things means a physical object, an action or idea, a situation or activity.

The goal of IoT is to extend to internet connectivity from standard devices like computer, mobile, tablet to relatively dumb devices like a toaster.

Internet of Things (IoT) is a network of physical objects or people called "things" that are embedded with software, electronics, network, and sensors that allows these objects to collect and exchange data.

Or

The Internet of Things, or IoT, refers to the billions of physical devices around the world that are now connected to the internet, all collecting and sharing data.

Or

IOT means a network of physical things sending , receiving or communicating information using the internet or other communication technologies and network just as the computer, tablets & mobiles do , & thus enabling the monitoring , coordinating or controlling process across the Internet or another data network.

Benefits of IoT:

Since IoT allows devices to be controlled remotely across the internet, thus it created opportunities to directly connect & integrate the physical world to the computer-based systems using sensors and internet. The interconnection of these multiple embedded devices will be resulting in automation in nearly all fields and also enabling advanced applications. This is

resulting in improved accuracy, efficiency and economic benefit with reduced human intervention.

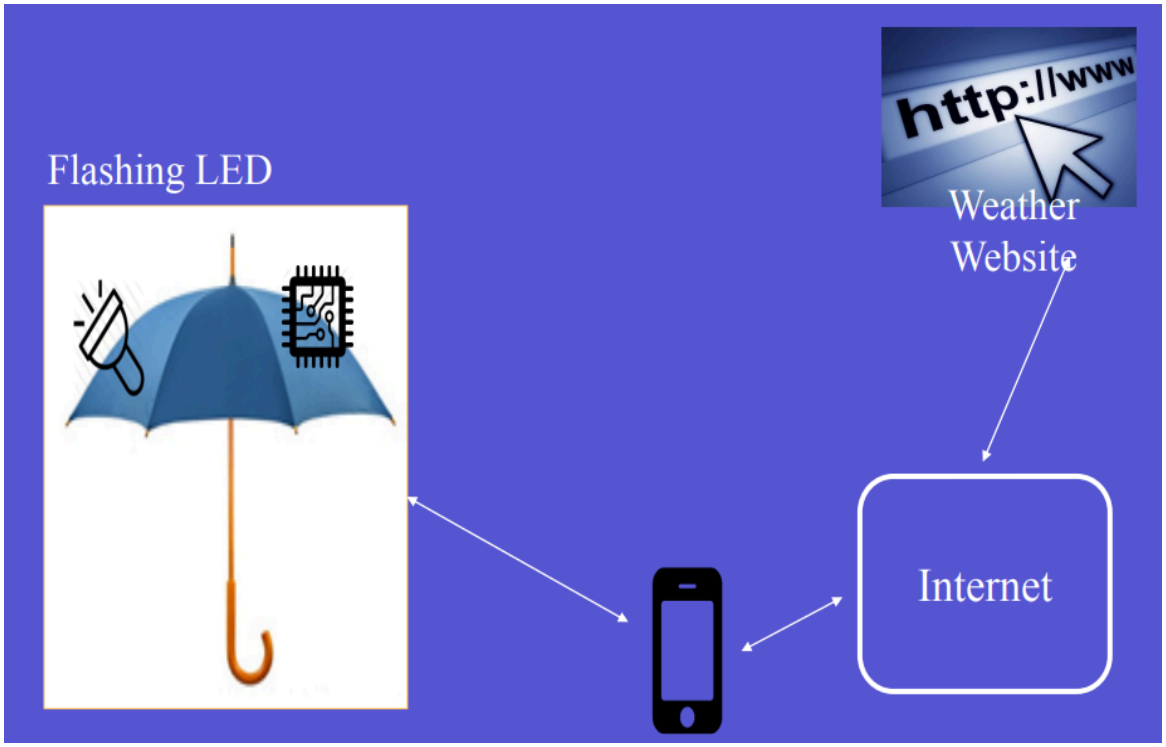
The major benefits of IoT are: Improved Customer Engagement ,Technical Optimization ,

Reduced Waste.

IOT Vision: the vision of IOT is where things (wearable watches, alarm, clocks, home devices, surrounding objects) become 'smart' and function like living entities by sensing computing and communicating systems. A vision where embedded devices interact with remote objects or persons through connectivity, for examples, using Internet or Near Field Communication or other technologies.

Example of IOT:

Smart Umbrella:



The umbrella, embedded with a circuit for the purpose of computing & 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 reminder can be distinguished using differently coloured LED flashes such as red flashes for hot & sunny days , yellow flashes for rainy days.

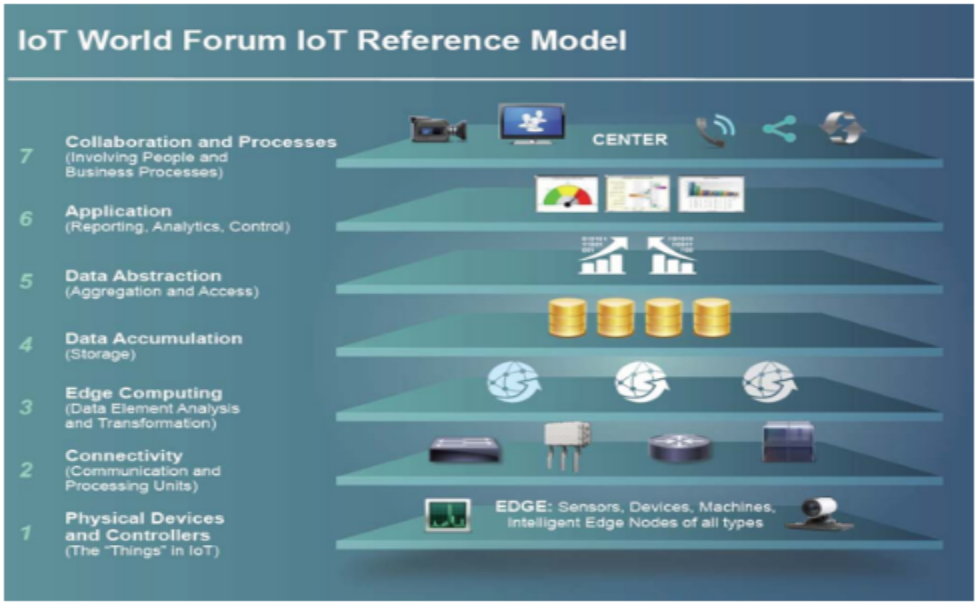
A reminder can be sent to the owner's mobile at a pre-set time before leaving for office using NFC, Bluetooth or SMS technologies.

The message can be i) protect yourself from rain, it is going to rain

ii) protect yourself from the sun, it is going to be hot & sunny. Know the owner can decide to carry or not to carry the umbrella.

2. IOT ARCHITECTURAL VIEW:

IoT solutions have become a regular part of our lives. From the smartwatch on your wrist to industrial enterprises, connected devices are everywhere. Having *things* work for us is no longer sci-fi fantasy. You tap the screen of your smart phone or say a word, and get immediate results. A door automatically opens, a coffee machine starts grinding beans to make a perfect cup of espresso while you receive analytical reports based on fresh data from sensors miles away.
But between your command and tasks fulfilled, there lies a large and mostly invisible infrastructure, that involves multiple elements and interactions.
The IoT Architecture generally comprises of 7 layears



IOT World Forum architectural committee published 7-layer IOT architectural reference model in 2014. This committee was led by cisco, IBM, Rockwell automation, & others. In this architecture edge computing, data storage & access were included.

Layer 1: Physical devices & control layer: which converts analog signals into digital data and vice versa. The is the initial stage of any IoT system consists of wide range of "things" or endpoint devices that act as a bridge between the real and digital worlds. They vary in form and size, from tiny silicon chips to large vehicles. this layer consisting of the "things" themselves and the sensors, machines, actuators and Edge Node devices. it gathers the data from environment & surroundings.

Sensors such as probes, gauges, meters, and others. They collect physical parameters like temperature or humidity, turn them into electrical signals, and send them to the IoT system. IoT sensors are typically small and consume little power.

Actuators, translating electrical signals from the IoT system into physical actions. Actuators are used in motor controllers, lasers, robotic arms.

Machines and devices connected to sensors and actuators or having them as integral parts.

It's important to note that the architecture puts no restriction on the scope of its components or their location. The edge-side layer can include just a few "things" physically placed in one room or myriads of sensors and devices distributed across the world.

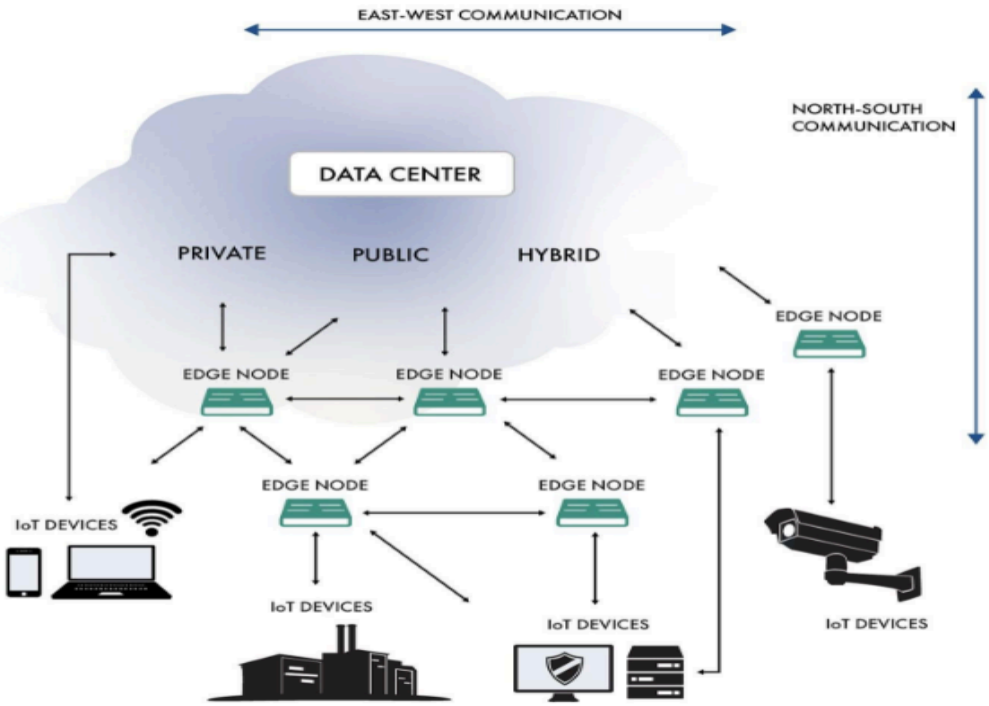
Layer 2: Connectivity layer: enabling data transmission

The second level is used for communications across devices, networks, and cloud services. This layer includes the mapping of field data to the logical and physical technologies used as well as the backhaul to the on premise or cloud and the next layer, Edge Computing.

This layer is responsible for reliable communication & transmission of data between the devices & networks. The connectivity between the physical layer and the cloud is achieved by using a single solution or multiple technologies, depending on the need. It may be wired (Ethernet, NFC, LPWAN (Low-power Wide-area Network) or wireless (WiFi, Bluetooth) technologies.

Layer 3: Edge or fog computing layer: reducing system latency

It involves data element analysis & transformation of data before sending data to the centers. This level is essential for enabling IoT systems to meet the speed, security, and scale requirements of the 5th generation mobile network or 5G.this layer interfaces the data and control plains to the higher layers of cloud. the idea behind edge or fog computing is to process and store information as early and as close to its sources as possible. This approach allows for analyzing and transforming high volumes of real-time data locally, at the edge of the networks. Thus, you save the time and other resources that otherwise would be needed to send all data to cloud services. The result is reduced system latency that leads to real-time responses and enhanced performance.



For ex sensors deployed in a particular location may sense roughly some values. instead of sending all the similar data to the cloud, sensed values can be aggregated like taking average or some operation are performing on that sensor values & than can be send to the cloud for storage. Due to that we can reduce the data volume & redundancy of data at cloud. So this allows the data is reformatted or decoded.

Edge computing occurs on gateways, local servers, or other edge nodes scattered across the network. At this level, data can be:

- evaluated to determine if it needs further processing at higher levels,
- formatted for further processing,
- decoded,
- filtered, and
- redirected to an additional destination

Layer 4 & 5: Processing layer: making raw data useful

The processing layer accumulates, stores, and processes data that comes from the previous layer. All these tasks are commonly handled via IoT platforms and include two major stages.

1. Data accumulation stage
2. Data abstraction stage

Data accumulation stage: it is essential to provide incoming data storage for subsequent processing, normalization, integration, and preparation for upstream applications. While part of the overall "data lake" architecture, this layer of the architecture serves the intermediate storage of incoming storage and outgoing traffic queued for delivery to lower layers. The data accumulation component stage works as a transit hub between event-based data generation and query-based data consumption. The total goal is to sort out a large amount of diverse data and store it in the most efficient way.

It also converts the event based data to the query based data. Event based data means when ever some event is occurring for ex when ever fire is detected by fire sensor, that data is stroing at the cloud. When that data is stored at cloud it becomes historical data. In future if you want to predict anything than you can use this historical data .

Data abstraction stage: this layer maintains consistency & consolidates data at one place by using aggregation & data access. Here, data preparation is finalized so that consumer applications can use it to generate insights. In data abstraction layer sense the data, collecting data from multiple sources, both IoT sensors or measurements, reconciling multiple data formats; aggregating data in one and flows for upstream processing.

Similarly, data collected at the application layer is reformatted here for sending to the physical level so that devices can "understand" it.

Layer 6: Application Layer: addressing business requirements delivering solutions like data analytics, reporting, and device control to end users.

At this layer, information is analyzed by software to give answers to key business questions. There are hundreds of IoT applications that vary in complexity and function, using different technology stacks and operating systems. by using some applications we can analyze the data. This application is varying on data & depends on business needs. For ex: aarogya sathu app.

Layer 7: **Collaboration and Proces**ses:

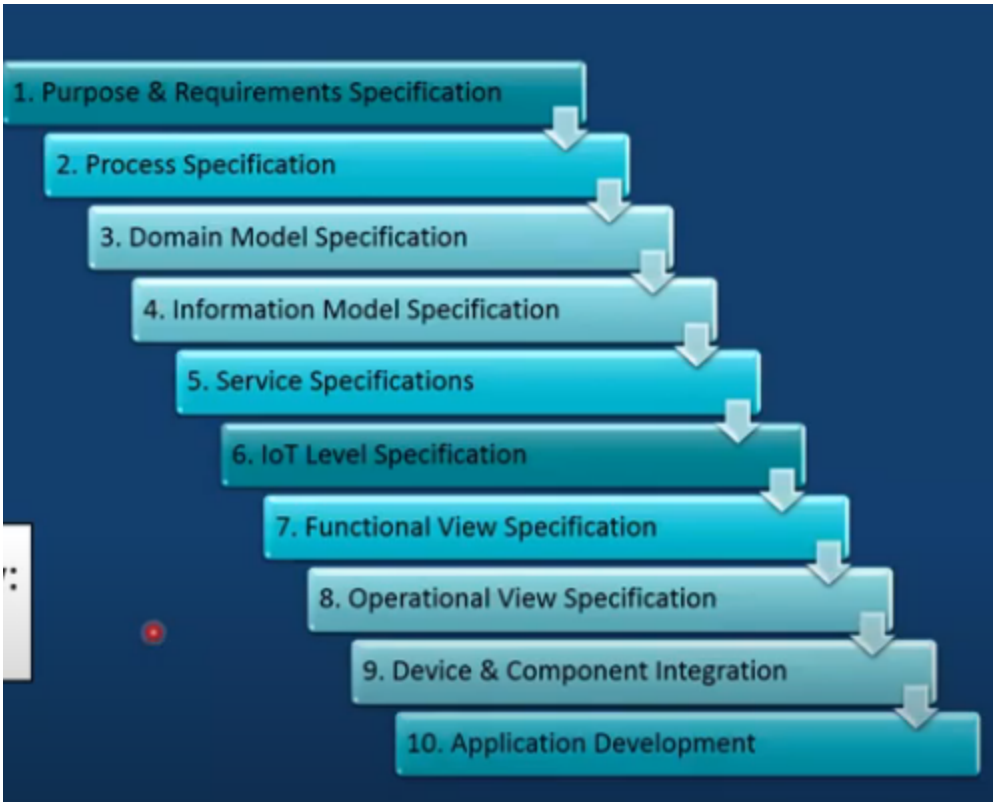
At this layer, application processing is presented to users, and data processed at lower layers is integrated in to business applications. This layer is about human interaction with all of the layers of the IoT system and where economic value is delivered. The challenge at this layer is to effectively leverage the value of IoT and the layers of infrastructure and services below and leverage this into economic growth, business optimization and/or social good.

3. IOT Design methodology:

IoT Design Methodology that includes:

1**. Purpose & Requirements Specification:**

The first step in IoT system design methodology is to define the purpose and requirements of the system. In this step, the system purpose, behavior and requirements (such as data collection requirements, data analysis requirements, system management requirements, data privacy and security requirements, user interface requirements, ...) are captured.



System Management Requirements: system should remotely provide monitoring and control functions

Data Analysis Requirements: The System should perform local analysis of the data.

Application Deployment Requirement : Deployed locally on device, but acts remotely without manual intervention.

Security: Authentication to Use the system must be available

2.Process Specification:

In this step, the use cases of the IoT system are formally described based on and derived from the purpose and requirement specifications.

3. Domain Model Specification:

Describes the main concepts, entities and objects in the domain of IoT system to be designed • It defines the attributes of the objects and relationships between them • . Domain model provides an abstract representation of the concepts, objects and entities in the IoT domain, independent of any specific technology or platform. With the domain model, the IoT system designers can get an understanding of the IoT domain for which the system is to be designed.

4. Information Model Specification:

Information Model defines the structure of all the information in the IoT system, for example, attributes of Virtual Entities, relations, etc. Information model does not describe the specifics of how the information is represented or stored. To define the information model, we first list the Virtual Entities defined in the Domain Model. Information model adds more details to the Virtual Entities by defining their attributes and relations.

5. Service Specifications:

Service specifications define the services in the IoT system, service types, service inputs/output, service endpoints, service schedules, service preconditions and service effects.

6. IoT Level Specification:

define the IoT level for the system. we defined 7 IoT deployment levels

7. Functional View Specification:

The Functional View (FV) defines the functions of the IoT systems grouped into various Functional Groups (FGs). Each Functional Group either provides functionalities for interacting with instances of concepts defined in the Domain Model or provides information related to these concepts.

8.Operational View Specification:

In this step, various options pertaining to the IoT system deployment and operation are defined, such as, service hosting options, storage options, device options, application hosting options, etc

9. Device & Component Integration:

The ninth step in the IoT design methodology is the integration of the devices and components.

10. Application Development:

The final step in the IoT design methodology is to develop the IoT application.

IoT Device Capabilities:

Each IoT device provides one or more capabilities—features or functions—it can use on its own 365 or in conjunction with other IoT and non-IoT devices to achieve one or more goals.

Transducer capabilities interact with the physical world and serve as the edge between digital and physical environments. Transducer capabilities provide the ability for computing devices to

interact directly with physical entities of interest. Every IoT device has at least one transducer capability. The two types of transducer capabilities are:

Sensing: the ability to provide an observation of an aspect of the physical world in the form of measurement data. Examples include temperature measurement, computerized tomography scans (radiographic imaging), optical sensing, and audio sensing.

Actuating: the ability to change something in the physical world. Examples of actuating capabilities include heating coils, cardiac electric shock delivery, electronic door locks, unmanned aerial vehicle operation, servo motors, and robotic arms.

• Data capabilities are typical digital computing functions involving data: data storing and data processing.

• Interface capabilities enable device interactions (e.g., device-to-device communications, human-to-device communications). The types of interface capabilities are:

Application interface: the ability for other computing devices to communicate with an IoT device through an IoT device application. An example of an application interface capability is an application programming interface (API).

Human user interface: the ability for an IoT device and people to communicate directly with each other. Examples of human user interface capabilities include keyboards, mice, microphones, cameras, scanners, monitors, touch screens, touchpads, speakers, and haptic devices.

Network interface: the ability to interface with a communication network for the purpose of communicating data to or from an IoT device—in other words, to use a communication network. A network interface capability includes both hardware and software (e.g., a network interface card and the software implementation of the networking protocol that uses the card). Examples of network interface capabilities include Ethernet, Wi-Fi, Bluetooth, Long-Term Evolution (LTE), and ZigBee. Every IoT device has at least one enabled network interface capability and may have more than one.

• Supporting capabilities provide functionality that supports the other IoT capabilities. Examples are device management, cyber security, and privacy capabilities.

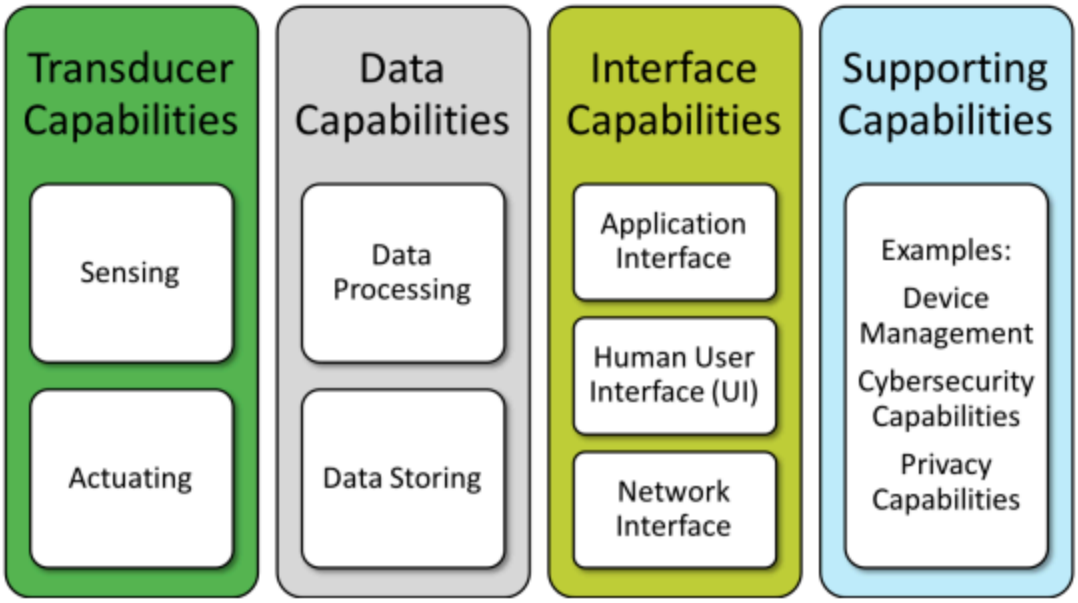
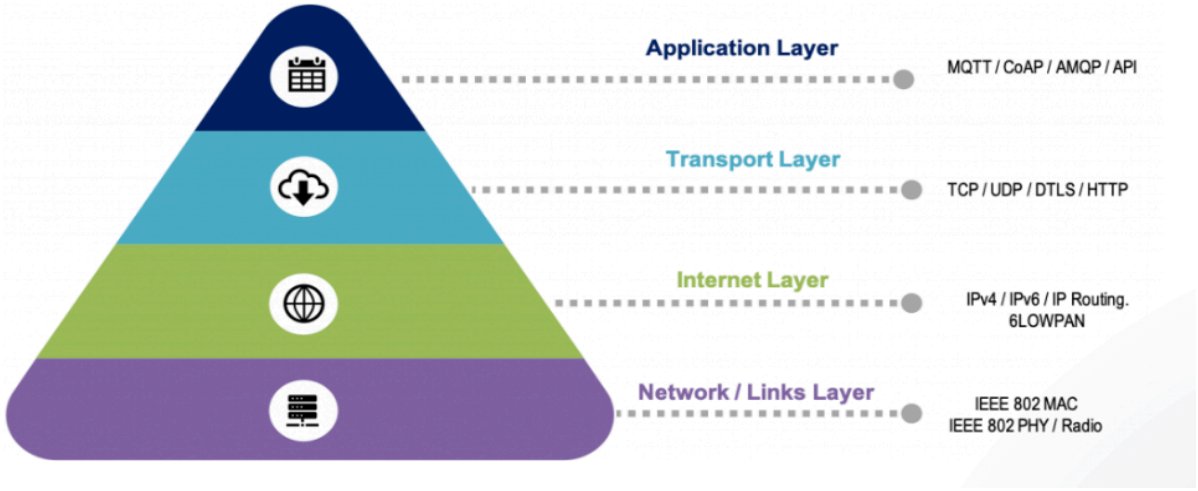


Figure 2: IoT Device Capabilities Potentially Affecting Cybersecurity and Privacy Risk

4.NETWORKS USED IN IOT:

IoT Network refers to the communication technologies used by Internet of Things (IoT) devices to share or spread the data to other device or interfaces available within reachable distance. There are various types of IoT networks available for IoT devices to communicate. It is critical to choose proper networking protocol for given requirements.



Here is a structure that pictures the network layering in IoT technology.

Link Layer(Ethernet):

It determines how the:

-data is physically sent over the network's physical layer or medium.

- packets are coded and signaled by the hardware device over the medium to which host is attached.

Sr.No	Standard	Shared medium
1	802.3	Coaxial Cable…10BASE5
2	802.3.i	Copper Twisted pair …..10BASE-T
3	802.3.j	Fiber Optic……10BASE-F
4	802.3.ae	Fiber…..10Gbits/s

Data Rates are provided from 10Gbit/s to 40Gb/s and higher

Link Layer(WiFi):

Sr.No	Standard	Operates in
1	802.11a	5 GHz band
2	802.11b and 802.11g	2.4 GHz band
3	802.11.n	2.4/5 GHz bands
4	802.11.ac	5 GHz band
5	802.11.ad	60 GHz band

.Collection of Wireless LAN communication standards

• Data Rates from 1Mb/s to 6.75 Gb/s

Link Layer(LR-WPAN):

Collection of standards for low-rate wireless personal area networks

• Basis for high level communication protocols such as Zigbee

• Data Rates from 40 Kb/s to 250 Kb/s

• Provide low-cost and low-speed communication for power constrained

Devices

Link Layer(2G/3G/4G-Mobile Communication):

Data Rates from 9.6Kb/s (for 2G) to up to 100Mb/s (for 4G)

GSM- Global System For Mobile

CDMA- Code Division Multiple Access

UMTS- Universal Mobile Telecommunications System

LTE- Long Term Evolution

Sr.No	Standard	Operates in
1	2G	GSM-CDMA
2	3G	UMTS and CDMA 2000
3	4G	LTE

Network /Internet Layer:

Responsible for sending of IP datagrams from source to destination network

• The datagram contain the source and destination addresses which are used to route them from source to destination across multiple networks.

• Performs the host addressing and packet routing

• Host identification is done using hierarchical IP addressing schemes such as IPV4 (internet version protocol 4) or IPV6 (internet version protocol 6)

IPV4

• Used to identify the devices on a network using hierarchical addressing scheme

• Uses 32-bit address scheme (232 or 4,294,967,296 addresses)

• IPV6

• Uses 128-bit address scheme (2128 or 3.4 x 1038 addresses)

• 6LoWPAN (IPV6 over Low power Wireless Personal Area Network)

• Used for devices with limited processing capability

• Operates in 2.4 GHz frequency

• Data Rates of 250 Kb/s

Transport Layer:

Provide end-to-end message transfer capability independent of the underlying network

• the message transfer capability can be setup on connections, either using handshakes (TCP) or with out handshake/ack (UDP).

• It provides functions such as error control, segmentation, flow-control and congestion control

TCP(Transmission Control Protocol):

It is used by web browsers , email programs and file transfer .

• Connection Oriented

• Ensures Reliable transmission

• Provides Error Detection Capability to ensure no duplicacy of packets and retransmit lost packets

• Flow Control capability to ensure the sending data rate is not too high for the receiver process

• Congestion control capability helps in avoiding congestion which leads to degradation of N/W Performance

UDP(User Datagram Protocol):

User Datagram Protocol

• Connection-less

• Does not ensures Reliable transmission

• Does not do connection before transmitting

• Does not provide guaranteed delivery, ordering of messages and duplicate elimination

• Transaction oriented and stateless (time sensitive applications- very small data need to exchange)

Application Layer:

How the applications interface with the lower layer protocols to send the data over network.

Port numbers are used for application addressing (HTTP-80, SSH-22)

• Includes commands such as GET,PUT, POST, HEAD, OPTIONS, TRACE..etc

• Follows a request-response model (where a client sends request to server using HTTP commands)

• Uses Universal Resource Identifiers (URIs) to identify HTTP resources

CoAP:

• Constrained Application Protocol

Used for Machine to machine (M2M) applications meant for constrained devices and N/W's

• Web transfer protocol for IoT and uses request-response model (it runs on top of UDP)

• Uses client –server architecture (using connection less datagrams)

• Supports methods such as GET,POST, PUT and DELETE

WebSocket:

Allows full-duplex communication over single socket (client to server)

• Based on TCP

• Client can be a browser, IoT device or mobile application

MQTT:

• Message Queue Telemetry Transport , light-weight messaging protocol

• Based on publish-subscribe model

• Well suited for constrained environments where devices have limited processing, low memory and N/W bandwidth requirement.

XMPP:

Extensible messaging and presence protocol

• For Real time communication and streaming XML data between N/W entities

• Used for Applications such as messaging, gaming, Multi-party chat and voice/video calls.

• Decentralized protocol and uses client server architecture.

• It supports both client to server and server to server communication paths.

DDS:

Data Distribution service is a data-centric middleware standard for

device-to-device or machine-to-machine communication.

• Publish subscribe model where publishers create topics to which subscribers can use.

• Provides Quality-of-service control and configurable reliability

AMQP:

Advanced Messaging Queuing Protocol used for business messaging.

• Supports both point-to-point and publisher/subscriber models, routing and queuing.

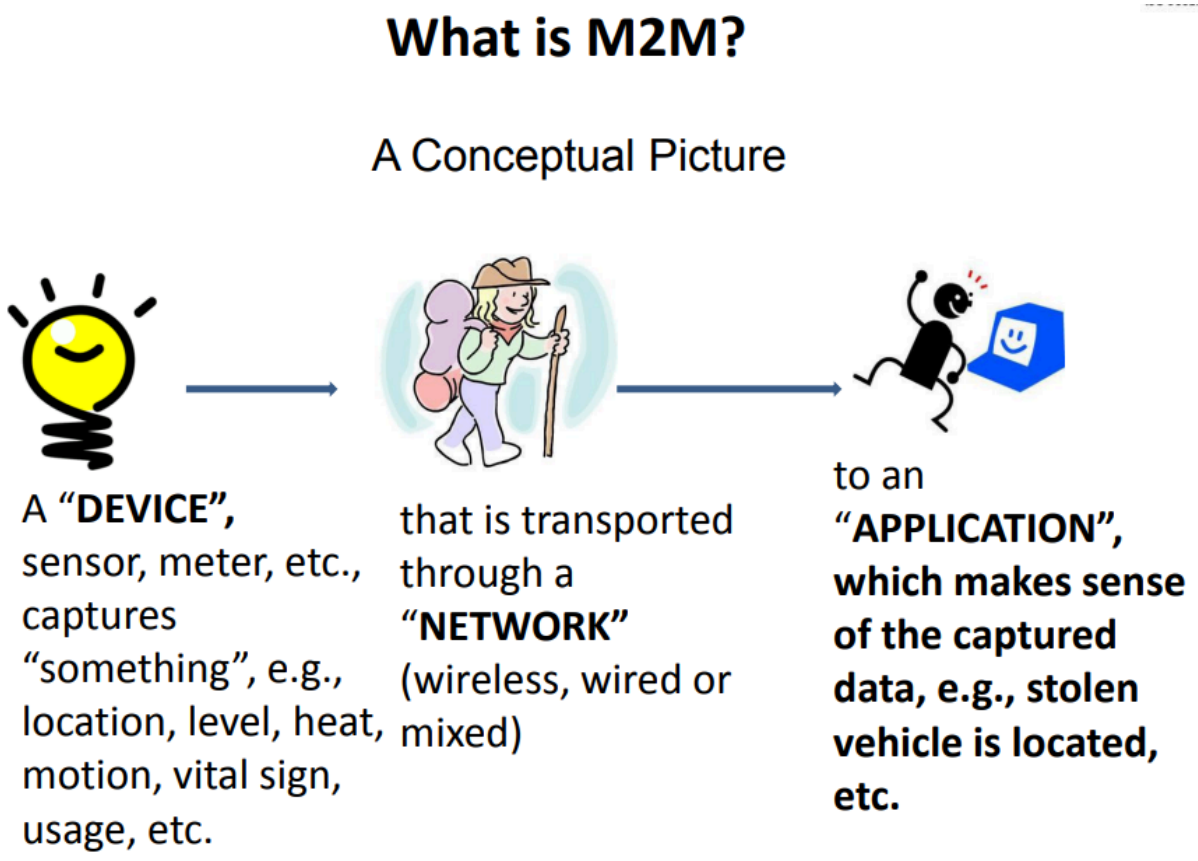
• Broker here receives messages from publishers and route them over connections to consumers through messaging queues

M2M COMMUNICATION:

Machine-to-machine communication, or M2M, is two or more machines "communicating," or exchanging data, without human interaction. M2M refers to the process of communication of a physical objects or device at machine with other of the same type, mostly for monitoring but also for control purpose. Each machine in M2M system uses a device such as sensors, rfid etc. The Device senses the data or status of the machine , that translate the sensed data into a meaningful information.

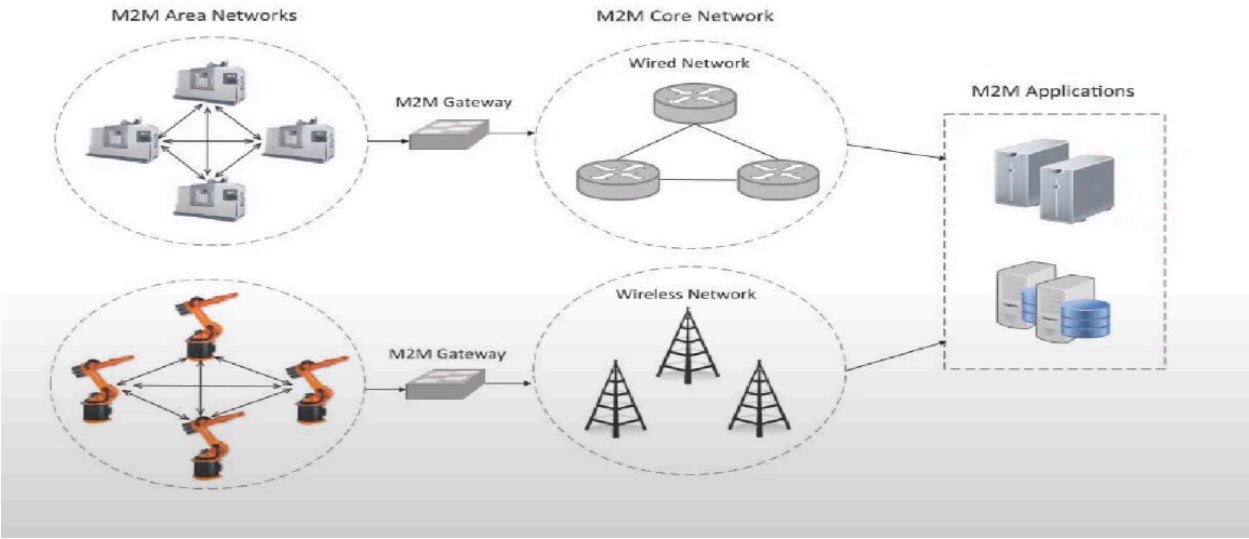
Communication in M2M may be wired or wireless systems. The communication Protocols are 6LowPAN, LWM2M, MQTT, XMPP. Each communication device is assigned 48-bits Ipv6 addresses

Examples of M2M communication in this case would be vending machines sending out inventory information or ATM machines getting authorization to despense cash.



M2M can refer to any two machines—wired or wireless—communicating with one another.

Machine-to-machine (M2M) communication allows machines and devices to pass along small amounts of information to other machines.



M2M refers to networking of machines for the purpose of remote monitoring and control and data exchange. As in above fig M2M system consists of M2M area networks, communication network and application domain. An M2M area network comprises of machines which have embedded hardware modules for sensing , actuation and communication. Various communication protocols can be used for M2M local area networks such as Zigbee, Bluetooth, m-bus, wireless m-bus, plc etc. these communication protocols provide connectivity between M2M nodes within an M2M area network. The communication network provides connectivity to remote M2M area networks. The communication network can use either wired or wireless networks. While the M2M area network use non-ip based communication protocols, the communication network uses ip based networks. Since to translate from non-ip to ip based protocols , M2M gateways are used.

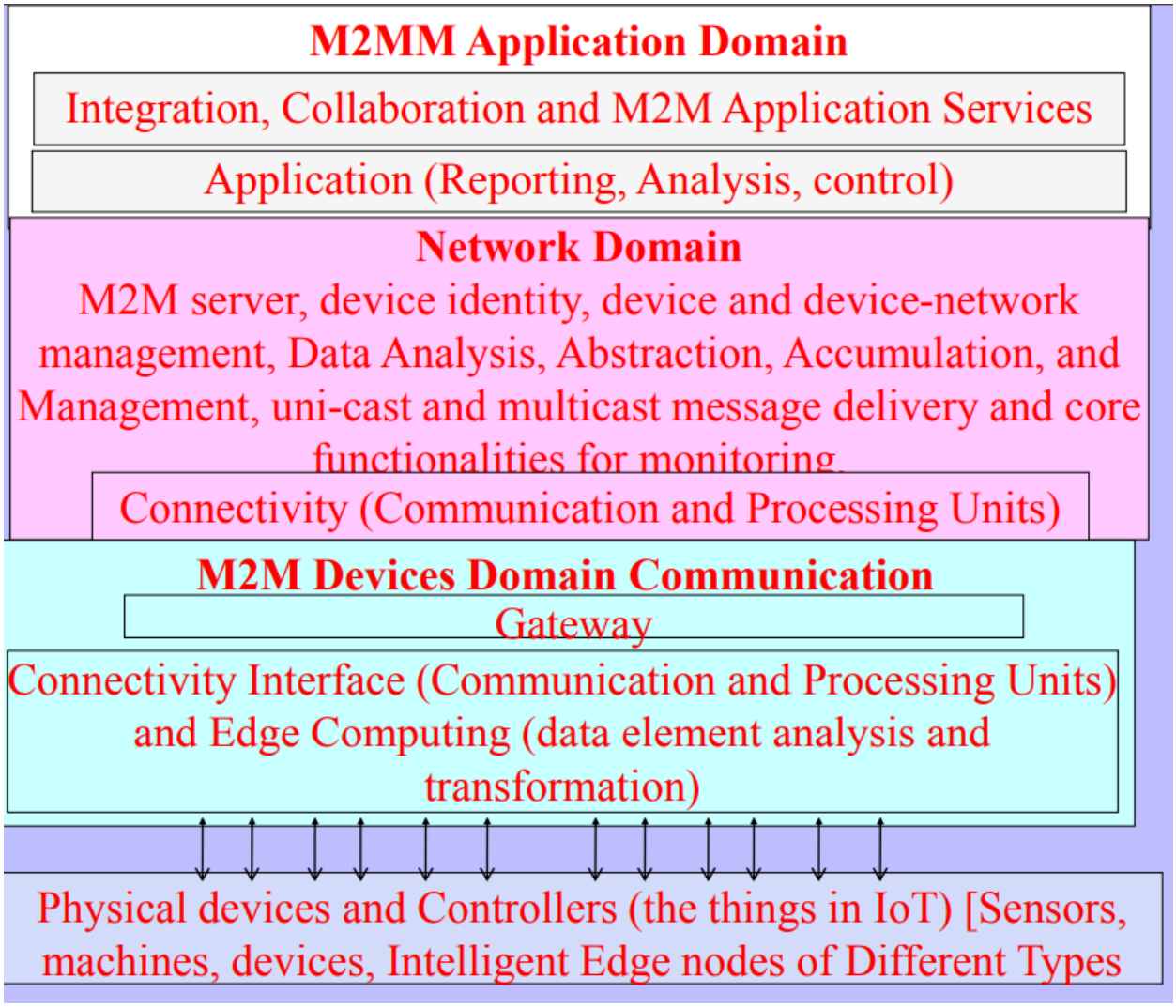
M2M Architecture:

M2M architecture consists of Three domains

• M2M Device domain,

• M2M network

• M2M Application domain



M2M Devices Domain Communication: Device domain Provide connectivity between M2M Devices and M2M Gateways, e.g. personal area network.

it consists of three entities: physical devices, communication interface and gateway. Sensors and communication devices are the endpoints of M2M applications. Sensors or devices sense the data

from environment and send to the gateway via connectivity interface. Generally, devices can connect directly to an operator's network, or they will probably interconnect using WPAN technologies such as ZigBee or Bluetooth. communication interface is a port or a subsystem, which receives the input from one end & sends the received data to another. The gateway module provides control & localization service for data collection. Gateways and routers are the endpoints of the operator's network in scenarios where sensors and M2M devices do not connect directly to the network.

M2M network domain: It provides the communications between the M2M Gateway(s) and M2M application(s), it uses wired or wireless networks such as LTE, WiMAX, Satillite communication and WLAN.

It consists of M2M server, device identity, device and device-network management, Data Analysis, Abstraction, Accumulation, and Management similar to IOT level. (connect+collect+assemble+analyse).

M2M Application domain: consists of applications for services, monitoring, analysis & controlling of devices networks. Here data goes through various app. Services & is used by the specific business processing engineer.

DIFFERENCE BETWEEN M2M and IoT:

Basis	IoT	M2M
Abbreviation	Internet of Things	Machine to Machine
Communication	IoT sensors automation	Communicates directly between machines
Connection	The connection is via using various communication types	Point to Point Connection
Communication protocols	HTTP, Ftp, Telnet, etc are used	Communication technology techniques and traditional protocols are used.
Intelligence	Objects are responsible for decision making	Observation of some degree of intelligence
Technology	Hardware and Software based technology	Hardware-based technology

Basis	IoT	M2M
Data Delivery	Depending on Internet protocol	Devices can be connected through mobile or other networks
Internet Connection	Active Internet connection is required	Devices do not rely on internet connection
Scope	Many users can connect at a time over the Internet	Communicate with a single machine at a time
Business Type	B2C(Business to Customers) and B2B(Business to Business)	Only B2B(Business to Business) is used
Open API support	IoT supports open API Integrations	M2M does not support open API
Data Sharing	Data is shared with applications that tend to improve the end-user experience	Data is shared with the communication parties themselves.

Fundamental of IoT's:

IOT connects multiple devices at a time to the internet there by facilitating man to machine & machine to machine interactions. IOT System is not limited to a particular field, but has applications in home automation, vehicle automation , factory line automation, healthcare etc.

IOT devices:

A device is a hardware unit that can sense aspects of its environment and/or actuate, i.e. perform tasks in its environment.

Devices are grouped into two categories

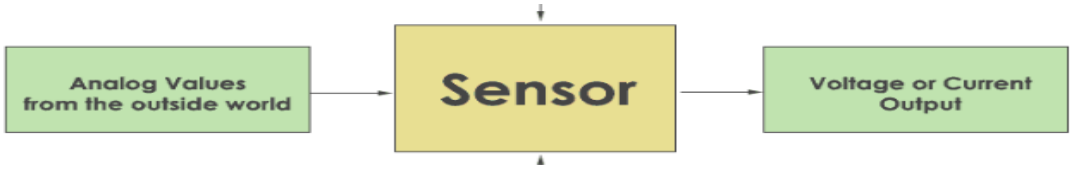
1. **Basic Devices:**
2. **Advanced Devices:**

Basic Devices: Devices that only provide the basic services of sensor readings and/or actuation tasks, and in some cases limited support for user interaction. LAN communication is supported via wired or wireless technology, thus a gateway is needed to provide the WAN connection.

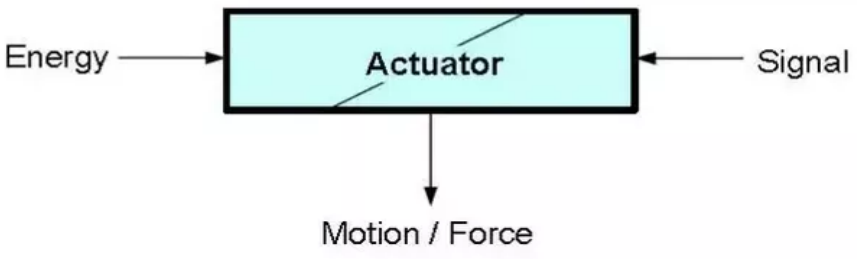
Advanced Devices: In this case the devices also host the application logic and a WAN connection. They may also feature device management and an execution environment for hosting multiple applications. Gateway devices are most likely to fall into this category.

there are many scenarios in which IoT can be employed and they all require different devices. Here, at the most **basic level,** we have **Sensors** and **actuators**.

Sensors is a devices that sense things i.e takes the input from environment, such as temperature, motion, particles, etc.& gives to the system by converting it.



actuators is a devices that act on things i.e it converts electrical signal into the physical events, such as switches , motors or rotors.



Advanced devices

An advanced device are the following:

• A powerful CPU or microcontroller with enough memory and storage to host advanced applications, such as a printer offering functions for copying, faxing, printing, and remote management.

• A more advanced user interface with, for example, display and advanced user input in the form of a keypad or touch screen.

• Video or other high bandwidth functions.

Internet of Things (IoT) Gateways:

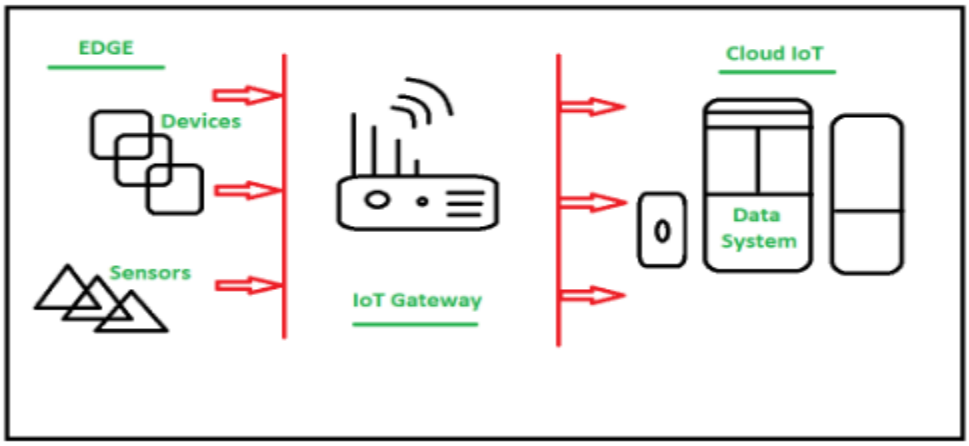
A Gateway is a hardware device that is used to connect two dissimilar type of networks. It allows us to send / receive data through the internet.

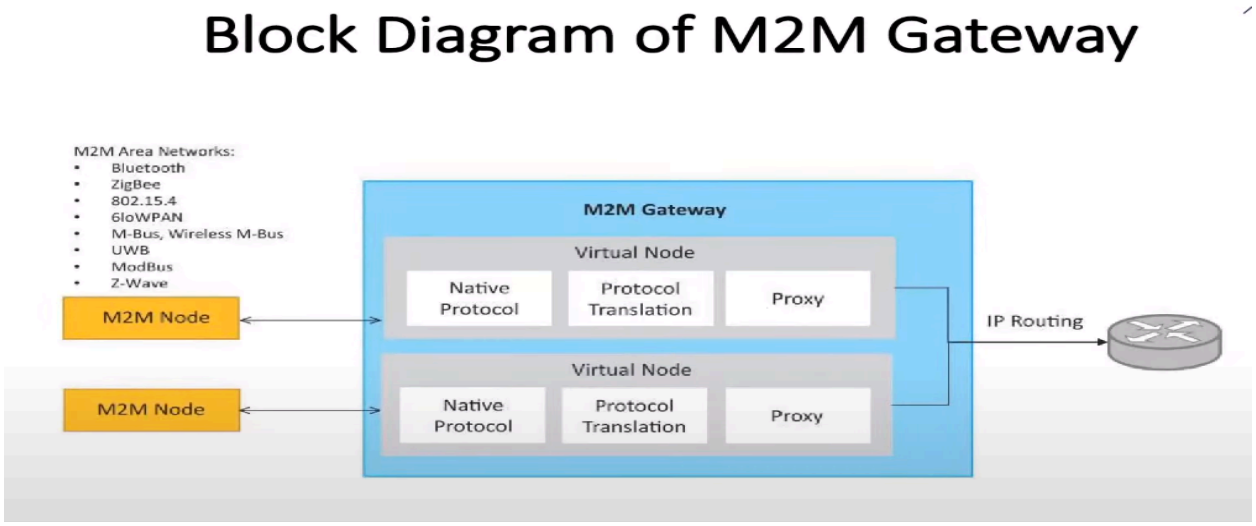
Gateway provides bridge between different communication technologies which means we can say that a Gateway acts as a medium to open up connection between cloud and controller(sensors / devices) in Internet of Things (IoT). By the help of gateways it is possible to establish device to device or device to cloud communication. A gateway can be a typical hardware device or software program.

It enables a connection between sensor network and Internet along with enabling IoT communication, it also performs many other tasks such as this IoT gateway performs protocol

translation, aggregating all data, local processing and filtering of data before sending it to cloud, locally storing data and autonomously controlling devices based on some inputted data, providing additional device security.

The below figure shows how IoT Gateways establish communication between sensors and cloud (Data System) :





The communication between the M2M nodes and the M2M gateways is based on the communication protocols which are native to the M2M area network. M2M gateway performs protocol translations to enable IP –connectivity for M2M area network. M2M gateway acts as a

proxy performing translation from/to native protocols to/from IP. With an M2M gateway, each node in an M2M area network appears as a virtualized node for external M2M area networks.

As IoT devices work with low power consumption(Battery power) in other words they are energy constrained so if they will directly communicate to cloud/internet it won't be effective in terms of power. So they communicate with Gateway first using short range wireless transmission modes/network like ZigBee, Bluetooth, etc as they consume less power or they can also be connected using long range like Cellular and WiFi etc.

Then Gateway links them to Internet/ cloud by converting data into a standard protocol like MQTT. using ethernet, WiFi/cellular or satellite connection. And in mostly Gateway is Mains powered unlike sensor nodes which are battery powered. In practice there are multiple Gateway devices.

Let's think about a simple IoT gateway, then our smartphone comes into picture as it can also work as a basic IoT gateway when we use multiple radio technologies like WiFi, Bluetooth, Cellular network of smart phone to work on any IoT project in sending and receiving data at that time this also acts as a basic IoT Gateway.

Key functionalities of IoT Gateway :

- Establishing communication bridge
- Provides additional security.
- Performs data aggregation.
- Pre processing and filtering of data.
- Provides local storage as a cache/ buffer.
- Data computing at edge level.
- Ability to manage entire device.
- Device diagnostics.
- Adding more functional capability.
- Verifying protocols.

Working of IoT Gateway :

1. Receives data from sensor network.
2. Performs Pre processing, filtering and cleaning on unfiltered data.
3. Transports into standard protocols for communication.

4. Sends data to cloud.
5. We can't access the internet without a gateway

IoT Gateways are key element of IoT infrastructure as Gateways establish connection for communication and also performs other task as described above. So IoT Gateway is one of most essential thing when we start think about an IoT ecosystem.

DATA MANAGEMENT:

Data management is a crucial aspect in the IOT. In the era of M2M, where billions of devices are interconnected and exchange all types of data, the volume of the generated data and the processes involved in the handling of those data becomes critical . so it is necessary to understand the challenges of data management.

Typical functions for data management include performing sensor readings and caching this data, as well as filtering, concentrating, and aggregating the data before transmitting it to back-end servers.

The data flow from the moment it is sensed (e.g. by a wireless sensor node) up to the moment it reaches the backend system has been processed manifold (and often redundantly), either to adjust its representation in order to be easily integrated by the diverse applications, or to compute on it in order to extract and associate it with respective business intelligence (e.g. business process affected, etc.).

In Figure 5.5, we see a number of data processing network points between the machine and the enterprise that act on the DataStream (or simply forwarding it) based on their end-application needs and existing context.

- Dealing with M2M data may be decomposed into several stages.
- Additionally, the degree of focus in each stage heavily depends on the actual usage requirements put upon the data as well as the infrastructure.

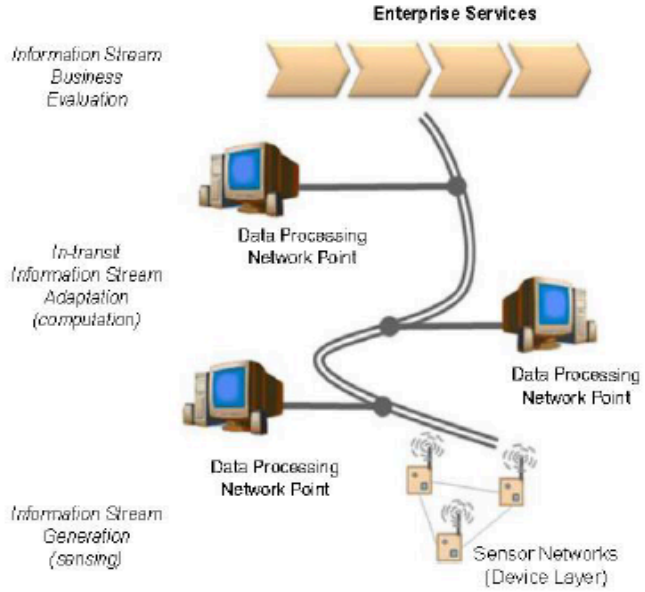


FIGURE 5.5

M2M data from point of generation to business assessment.

1.Data generation:

Data generation is the first stage within which data is generated actively or passively from the device, system, or as a result of its interactions.

☂ The sampling of data generation depends on the device and its capabilities as well as potentially the application needs .

Data generates at devices that later on, transfers to the internet through a gateway..

The data generates as follows

Passive device data: the data generates at the device or system, due to the result of interactions. A passive device does not have its own power source. An external source helps such a device to generate & send data. Ex: RFID tag

Active device data: the data generates at the device or system or following the result of interactions. A passive device has its own power source. Ex: Active RFID, streetlight sensor. an active device also has an associated microcontroller, memory & transceiver.

Event data: a device can generate data on an event only once. Ex: the event on darkness communicates a need for lighting up a group of streetlights.

Device real-time data: an ATM generates & communicates it to the server instantaneously through the internet. This initiates & enables online transactions processing (OLTP) in real time

2.Data acquisition:

Data acquisition deals with the collection of data (actively or passively) from the IOT or M2M device, system, or as a result of its interactions.

- The data acquisition systems usually communicate with distributed devices over wired or wireless links to acquire the needed data, and need to respect security, protocol, and application requirements.
- The nature of acquisition varies, e.g. it could be continuous monitoring, interval-poll, event-based, etc.

The data acquired at this stage (for non-closed local control loops) may also differ from the data actually generated. Ex: system can configure an umbrella device to acquire weather data from the internet weather service, once each working day in a week. (aggregated data).

- In simple scenarios, due to customized filters deployed at the device, a fraction of the generated data may be communicated.

3.Data validation:

Data acquired must be checked for correctness and meaningfulness within the specific operating context. Data validation software do the validation checks on the acquired data.

- This is usually done based on rules, semantic annotations, or other logic.
- The acquired data may not conform to expectations and data may be intentionally or unintentionally corrupted during transmission, altered, or not make sense in the business context.
- the applications or services depends on valid data. Then only the analytics, predictions, prescriptions, diagnosis & decisions can be acceptable.

4.Data storage:

The data generated by M2M interactions is what is commonly referred to as “Big Data.”

- Machines generate an incredible amount of information that is captured and needs to be stored for further processing.
- As this is proving challenging due to the size of information, a balance between its business usage vs. storage needs to be considered; that is, only the fraction of the data relevant to a business need may be stored for future reference.
- However, one has to carefully consider what the value of such data is to business not only in current processes, but also potentially other directions that may be followed in the future by the

company as different assessments of the same data may provide other, hidden competitive advantages in the future.

- Due to the massive amounts of M2M data, as well as their envisioned processing (e.g. searching), specialized technologies such as massively parallel processing DBs, distributed file systems, cloud computing platforms, etc. are needed.

5.Data processing

- Data processing enables working with the data that is either at rest (already stored) or is in-motion (e.g. stream data).
- The scope of this processing is to operate on the data at a low level and "enhance" them for future needs.
- Typical examples include data adjustment during which it might be necessary to normalize data, introduce an estimate for a value that is missing, re-order incoming data by adjusting timestamps, etc.

6.Data analysis

- Data available in the repositories can be subjected to analysis with the aim to obtain the information they encapsulate and use it for supporting decision-making processes.
- The analysis of data at this stage heavily depends on the domain and the context of the data.
- For instance, business intelligence tools process the data with a focus on the aggregation and key performance indicator assessment.
- Data mining focuses on discovering knowledge, usually in conjunction with predictive goals.
- Statistics can also be used on the data to assess them quantitatively (descriptive statistics), find their main characteristics (exploratory data analysis), confirm a specific hypothesis (confirmatory data analysis), discover knowledge (data mining), and for machine learning, etc.
- This stage is the basis for any sophisticated applications that take advantage of the information hidden directly or indirectly on the data.

Business processes in IoT:

- It consists of a series of activities, which serves a particular result. BP is used when an enterprise has a number of interrelated processes which serve a particular goal. Which results enable sales, planning & production.
- The BP is a representation or process matrix or flowchart of a sequence of activities with interleaving decision points.
- IOT/M2M enables the devices, data in database for business processes. The data supports the process.
- For ex: streetlight control & processes
- Each group of streetlights sends data in real time through the gateways. The gateways connect to the internet. The control & management processes streetlights real time database & group databases.

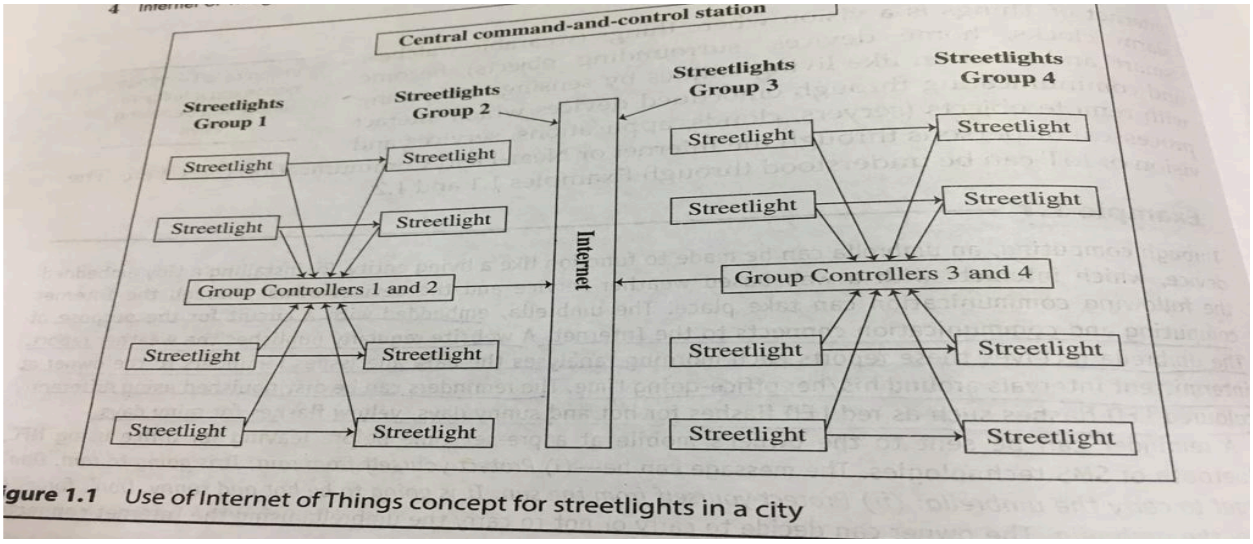


Figure 1.1 Use of Internet of Things concept for streetlights in a city

Business intelligence:

- it is a process which enables a business service to extract new facts & knowledge & then undertake better decisions.
- The new facts & knowledge follow from the earlier results of data processing, aggregation & then analyzing those results.

Distributed business process:

- Business processes need to be distributed. Distribution of processes reduces the complexity, communication costs, enables faster responses & smaller processing load at the central system.
- For ex: distribution of control process for each group of lights at the gateway itself reduces complexity, communication costs, faster responses & smaller processing load at the central system.
- DBPS(**Distributed business process System)** is a collection of logically interrelated business processes in an enterprise network. DBPS means a software system that manages the distributed BPs.
- DBPS exits as cooperation between the BPs in a transparent manner. Transparent means that each user within the system may access all of the process decisions within all of the processes as if they were a single business process.

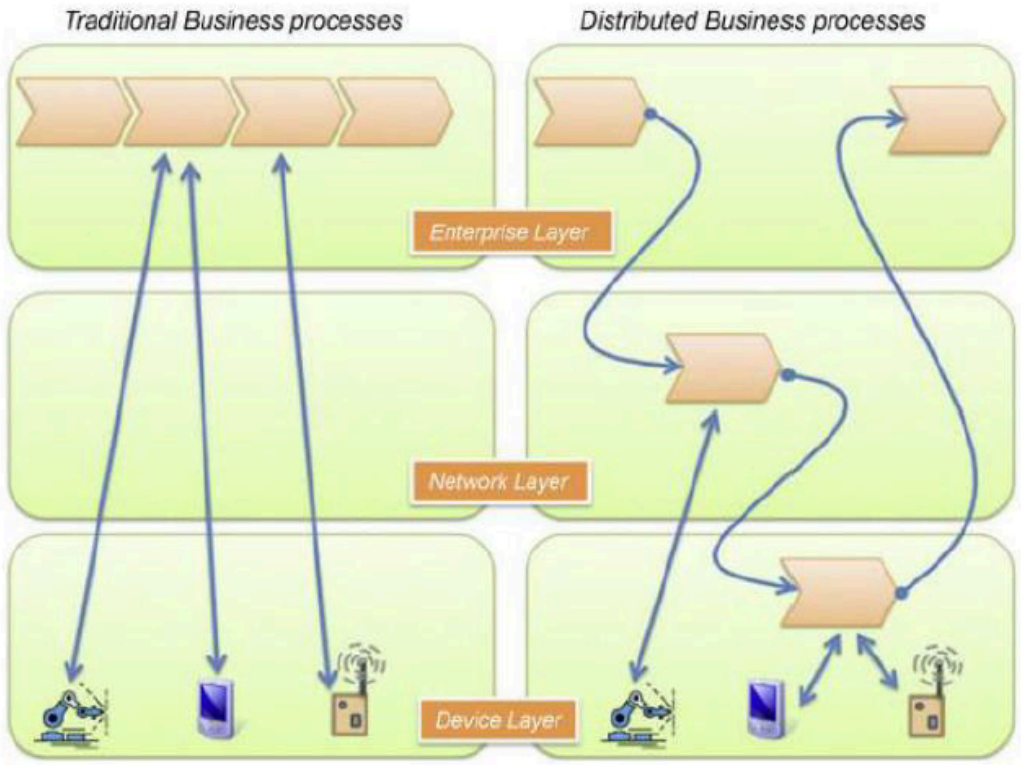


FIGURE 5.9

Distributed Business Processes in M2M era.

The first step is to minimize communication with enterprise systems to only what is relevant for business. With the increase

- in resources (e.g. computational capabilities) in the network, and especially on the devices themselves (more memory, multi-core CPUs, etc.), it makes sense not to host the intelligence and the computation required for it only on the enterprise side, but actually distribute it on the network, and even on the edge nodes (i.e. the devices themselves), as depicted on the right side of Figure 5.9.
- Partially outsourcing functionality traditionally residing in backend systems to the network itself and the edge nodes means we can realize distributed business processes whose sub-processes may execute outside the enterprise system.
- As devices are capable of computing, they can either realize the task of processing and evaluating business relevant information they generate by themselves or in clusters.
- Business processes can bind during execution of dynamic resources that they discover locally, and integrate them to better achieve their goals.

Everything as a service (XaaS):

An extensive number of modern digital services, products and tools are ordered over the Internet and delivered to users on demand, rather than provided via local channels within enterprises or specialized organizations. To describe this phenomenon, a special term was invented: **Everything-as-a-Service (XaaS)**.

Everything as a Service (XaaS):

Before only cloud computing technology was there and various cloud service providers were providing various cloud services to the customers. But now a new concept has been emerged i.e Everything as a Service (XaaS) means anything can now be a service with the help of cloud computing and remote accessing. Where cloud computing technologies provide different kinds of services over the web networks. In Everything as a Service various number of tools and technologies and services are provided to users as a service. Before XaaS and cloud services, companies have to buy licensed products and install them, had to all securities on their site and provide infrastructure for the business purposes. With XaaS, business is simplified as they have to pay for what they need. This Everything as a Service is also known as Anything as a Service. It is an extremely wide-ranging term that refers to any tools, applications, services, games, etc., which are delivered to your laptop or other device via the cloud, rather than obtained on-premises or in a physical format.

Examples of XaaS :

As XaaS stands for "Everything as a service", There are many examples. There are many varieties of cloud computing models like –

1. Software as a Service (SaaS)
2. Platform as a Service (PaaS)
3. Disaster Recovery as a Service (DRaaS)
4. Infrastructure as a service (IaaS)
5. Communication as a Service (CaaS)
6. Network as a Service(NaaS)
   Database as a Service (DBaaS)
7. Desktop as a Service (DaaS) etc.

SaaS provides many software applications like Google Apps, Microsoft Office 365. Similarly, PaaS offers AWS, Heroku, Apache Stratos, other sources relating application development and testing. IaaS helps to deploy and configure virtual machines and manage these remotely. IaaS also provide services to Azure and Google Computer Engine.

Everything as a Service Model Examples :

1. **Hardware as a Service (HaaS) –**
   Managed Service Providers (MSP) provide and install some hardware on customer's site on demand. Customer uses the hardware according to service level agreements. This model is very similar to IaaS as computing resources are present at MSP's site are provides to users substituted for physical hardware.
2. **Communication as a Service (CaaS) –**
   This model comprises solution of different communication like IM, VoIP, video conferencing application which are hosted in provider's cloud. Such method is cost-effective and reduces time expenses.
3. **Desktop as a Service (DaaS) –**
   DaaS provider mainly manages storing, security and backing up user data for the desktop apps. And a client can also work on PCs using third-party servers.
4. **Security as a Service (SECaaS) –**
   In this method provider integrates security services with company's infrastructure through internet which includes anti-virus software, authentication, encryption etc.

5. **Healthcare as a Service (HaaS) –**
   The healthcare industry has opted the model HaaS service through electronic medical records (EMR). IOT and other technologies has enhanced medical services like online consultations, health monitoring 24/7, medical service at doorstep e.g. lab sample collection from home etc.
6. **Transport as a Service (TaaS) –**
   Nowadays, there are numerous apps which helps in mobility and transport in modern society. The model is both convenient and ecological friendly e.g. Uber taxi services is planning to test flying taxis and self-driving planes in the future.
7. Saas:In this version, a provider hosts applications and software in the cloud and then offers them to consumers on a subscription basis. A good example of this would be Adobe, which offers a range of packages. Depending on the needs of the customer, they may choose to subscribe to one application, such as Adobe Photoshop, or to one of the various bundles of applications.

8. **PaaS**: With PaaS, the service provider delivers a platform to clients where they can host, run, manage, or develop applications without having to create or maintain their own on-premises or cloud infrastructure.

9. **IaaS**: IaaS, also known as cloud infrastructure services, consists of automated and easily scalable computing resources. This allows businesses access to all the IT infrastructure they need, whether on a temporary or long term basis. The favourable pricing of this service model offers the advantage to companies of not having to invest in hardware that may only be used a few times.

Benefits in XaaS :

- **Cost Saving –**
  When an organization uses XaaS then it helps in cost-cutting and simplify IT deployments.
- **Scalability –**
  XaaS can easily handle growing amount of works by providing required resources/service.
- **Accessibility –**
  It helps in easy accessing and improving accessibility as long as internet connection is there.

- **Faster Implementation –**
  It provides faster implementation time to various activities of organization.
- **Quick Modification –**
  It provides updates for modification as well as undergoes quick updating by providing quality services.
- **Better Security –**
  It contains improved security controls and configured to exact requirements of business.
- **Boost innovation –**
  While XaaS is used it Streamline the operations and free up resources for innovation.
- **Flexibility –**
  XaaS provides flexibility by using cloud services and multiple advanced approaches.

Disadvantages in XaaS :

- **Internet Breakage –**
  Internet breaks sometimes for XaaS service provider where there can also be issues in internet reliability, provisioning and managing the infrastructure resources.
- **Slowdown –**
  When too many clients are using same resources at the same time, the system can slow down.
- **Difficult in Troubleshoot –**
  XaaS can be a solution for IT staff in day-to-day operational headaches, but if anywhere problem occurs it is harder to troubleshoot it as in XaaS multiple services are included with various technologies and tools.
- **Change brings problem –**
  If XaaS provider discontinues a service or alters it gives impact to XaaS users.

Role of Cloud computing in IoT:

IoT and cloud computing complement each other. Both are working towards increasing the efficiency of everyday tasks. While IoT has penetrated mainstream technology and market place, it generates a massive amount of big data. Besides, cloud computing paves the way for this enormous data.

The data gathered by them interpret into meaningful information and pave the way for the advancement of IoT. Cloud services facilitate instantaneous databases, on-demand delivery of

computing infrastructure, storage. It also facilitates applications needed for the analysis and processing of data points generated through hundreds of IoT devices.

Based on the principles of agility and scalability, the cloud is acclaimed as an innovative technology across the globe. Cloud solutions can aid in the large-scale adoption of IoT initiatives.

Cloud is a centralized system that helps to deliver & transport data and various files across the internet to data centers. The different data & programmes can be accessed easily from the centralized cloud system. Cloud computing is an economic solution, as it does not require on-site-infrastructure for storage, processing & analytics. There are different types of cloud services available, including Microsoft azure cloud development, amazon web services etc. cloud computing helps to speed up the development process & cut down the development costs.

Cloud computing divided into front end & back end.

Front end provides the user to access data using internet browser or cloud computing software. Back end securely store the data.

Benefits of Cloud in IoT

Scalability

The scalability of cloud computing means that as your business grows, you are technolical & analytical capabilities can too.

Hosting your application on the cloud gives an unlimited room for scalability, which cannot be provided by the on-premise infrastructure. Scaling on the on-premise infrastructure may be very expensive as it would require buying more hardware, increased configurations, and more deployment time. When scaling on the cloud is less expensive as it just involves leasing more storage space. The cloud also offers flexibility, enabling you to scale up or down the number of IoT devices and applications that you can use.

Data Mobility

With the data stored and processed in the cloud server, it can be accessed from almost anywhere in the world, which also means that it won't be bound by any infrastructural or network limitations. Mobility is very essential when it comes to IoT projects requiring real-time

monitoring and management of connected devices.

Provide Security :

IoT devices collect all types of data, including sensitive data such as health, financial and personally identifiable information (PII). This data requires protection from privacy and integrity breaches by malicious actors. Cloud computing provides a secure storage environment for this data which is monitored all the time. The cloud also ensures regular updates to their platforms, firmware, and applications to eliminate known vulnerabilities.

Cost-Effectiveness

Large initial upfront investments and enhanced implementation risk in the case of an on-premise Internet of Things system can be discouraging. Adding to that, there is the issue of continuous costs of hardware maintenance and IT help. From the cloud prospect, things look better. Significantly diminished up-front costs and a flexible pricing plan based on pay per use encourage IoT-based businesses to switch to the cloud. Within this enterprise model, costs are easier to predict. You don't have to worry about hardware failure, which in case of on-premise Internet of Things systems may generate huge additional costs, apart from business losses resulting from service downtime.

SECURITY ACCEPTS IN IOT:

Some of the most required capabilities of a secure network are briefly discussed.

• **Resilience to attacks:** The system should be capable enough to recover itself in case if it crashes during data transmission. For an example, a server working in a multiuser environment, it must be intelligent and strong enough to protect itself from intruders or an eavesdropper. In the case, if it is down it would recover itself without intimation the users of its down status.

• **Data Authentication:** The data and the associated information must be authenticated. An authentication mechanism is used to allow data transmission from only authentic devices.

• **Access control:** Only authorized persons are provided access control. The system administrator must control access to the users by managing their usernames and passwords and by defining their access rights so that different users can access only relevant portion of the database or programs.

• Client privacy: The data and information should be in safe hands. Personal data should only be accessed by authorized person to maintain the client privacy. It means that no irrelevant

authenticated user from the system or any other type of client cannot have access to the private information of the client.

Identifying the right IoT security solution

With so many concerns surrounding IoT network security, you could be forgiven for wondering if it's worth implementing IoT at all. But the good news is that IoT and security don't have to be mutually exclusive. They can—and must—be fully integrated parts of the same solution.

At Bridgera, separating IoT and security issues is fundamental to our entire philosophy. When we develop a custom IoT solution, whether it's for a logistics use case or a healthcare implementation, we make a point of designing the risks out of the system. IoT security is an integral part of our IoT service enablement, not an add-on or an afterthought.

Do you already have an IoT solution in place that's running up against some of the IoT security challenges outlined above? Or maybe you're planning to build one but are worried about IoT security issues? Contact Bridgera today and schedule a free call with one of our IoT experts today. We'll work with you to develop a custom IoT solution that's cost-effective, full-featured and above all secure.