GOVT. POLYTECHNIC BALASORE

[LECTURE NOTES]

INTERNET OF THINGS
TH 4

DIPLOMA

6TH SEMESTER, E&TC ENGINEERING

PREPARED BY:-ER. PRAKASH CHNDRA DAS

SESSION : 2022(S)

DEPT. OF ELECTRONICS &TELECOMMUNICATION ENGINEERING

Th.4(ii)-INTERNET Of THINGS

Name of the Course: Diploma in **Electronics & Communication** Engineering			
Course code:		Semester	6th
Total Period:	60	Examination	3 hrs
Theory periods:	4P / week	Class Test:	20
Tutorial:		End SemesterExamination:	80
Maximum marks:	100		

(Elective)

A. RATIONALE:

Internet of Things and develop skills required to build real-life IoT based projects. The Internet of things describesthenetworkofphysicalobjects—-things—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the Internet. The goal behind the Internet of things is to have devices that self report in real-time, improving **efficiency** and bringing important information to the surface more quickly than a system depending on human intervention.Smart surveillance, automated transportation, smarter energy management systems, water distribution, urban security and environmental monitoring all are examples of **internet of things** applications for smartcities

B. OBJECTIVE:

After completion of this subject the student will be able to know:

1. Understand internet of Things and its hardware and software components
2. Interface I/O devices, sensors & communication modules
3. Remotely monitor data and control devices
4. Develop real life IoT based projects

C. Topic wise distribution ofperiods:

Sl. No.	Topics	Period
1	Introduction to IoT	08
2	Elements of IoT	10
3	IoT Application Development	14
4	Smart Technology	10
5	Smart TVs: Viewing in a Connected World	08
6	IoT Case Studies	10
	Total:	**60**

D. COURSECONTENTS:

1. Introduction to IoT

- 1.1 What isIoT.
- 1.2 ArchitecturalOverview,
- 1.3 Design principles and neededcapabilities,
- 1.4 IoT Applications, Sensing,Actuation,
- 1.5 Basics of Networking, M2M and IoTTechnology
- 1.6 Fundamentals- Devices andgateways,
- 1.7 Data management, Business processes inIoT,

1.8 Everything as aService(XaaS),
1.9 Role of Cloud in IoT, Security aspects inIoT.

2. Elements ofIoT

2.1 Hardware Components- Computing (Arduino,RaspberryPi),
2.2 Communication, Sensing, Actuation, I/Ointerfaces.
2.3 Software Components- Programming API's (using Python/Node.js/Arduino) forCommunication
2.4 Protocols-MQTT, ZigBee, Bluetooth, CoAP, UDP,TCP.

3. IoT ApplicationDevelopment

3.1 Solution framework for IoTapplications-
3.2 Implementation of Deviceintegration,
3.3 Data acquisition andintegration,
3.4 Device data storage- Unstructured data storage on cloud/localserver,
3.5 Authentication, authorization of devices.

4. Smart Technology

4.1 Understanding the IoT BigPicture
4.2 Building the Internet of Things
4.3 Understanding Smart Devices, BuildingBlocks
4.4 Understanding Network Connections
4.5 Understanding IPAddresses
4.6 Understanding cellular Network & MeshNetwork

5. Smart TVs: Viewing in a Connected World

5.1 What is Smart TV & itsuse
5.2 What is inside SmartTV
5.3 What a Smart TVdoes
5.4 Smart TV OperatingSystems
5.5 What is Smart TVSet-Top Devices
5.6 Integrating Smart TV in toIOT

6. IoT Case Studies

6.1 IoT case studies (anyone)
a. SmartHome
b. Smartcar
c. SmartCities
d. SmartDrones
6.2 Industrialautomation,

CHAPTER 1
INTRODUCTION OF IOT

WHAT IS IOT

The **Internet of things** (**IoT**) describes the network of physical objects—"things" or objects—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the Internet

Things have evolved due to the convergence of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded systems. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", including devices and appliances (such as lighting fixtures, thermostats, home security systems and cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smart phones and smart speakers. IoT can also be used in healthcare systems.

Definition:

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

Characteristics:

Dynamic & Self Adapting: IoT devices and systems may have the capability to dynamically adapt with the changing contexts and take actions based on their operating conditions, users context or sensedenvironment.

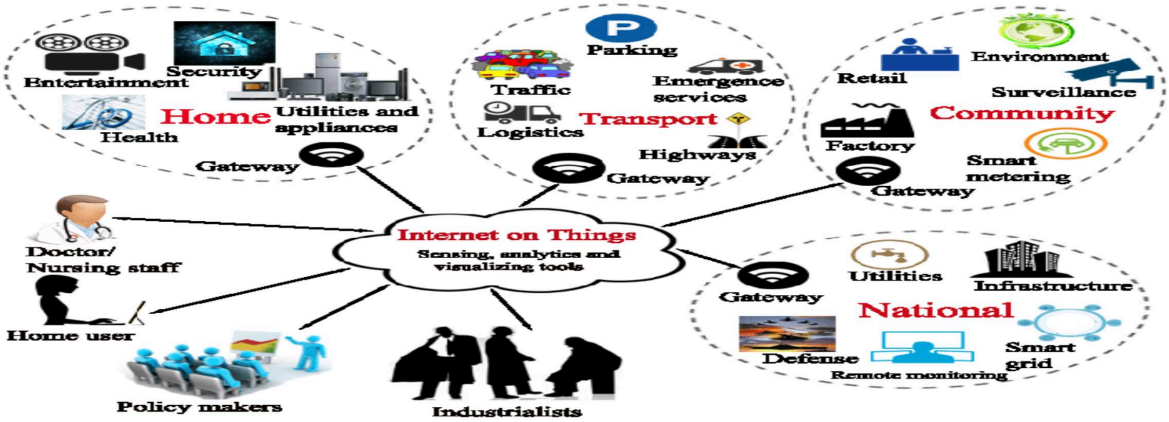
Eg: the surveillance system is adapting itself based on context and changing conditions.

Self-Configuring: allowing a large number of devices to work together to provide certain functionality.

Inter Operable Communication Protocols: support a number of interoperable communication protocols and can communicate with other devices and also with infrastructure.

Unique Identity: Each IoT device has a unique identity and a unique identifier(IP address).

Integrated into Information Network: that allow them to communicate and exchange data with



other devices and systems

ARCHITECTURAL OVERVIEW

What is the Internet of Things? - The concept of connecting any device (so long as it has an on/off switch) to the Internet and to other connected devices. The IoT is a giant network of connected things and people – all of which collect and share data about the way they are used and about the environment around them.

How does it work? Devices and objects with built in sensors are connected to an Internet of Things platform, which integrates data from the different devices and applies analytics to share the most valuable information with applications built to address specific needs.

These powerful IoT platforms can pinpoint exactly what information is useful and what can safely be ignored. This information can be used to detect patterns, make recommendations, and detect possible problems before they occur.

For example, if a car manufacturing business, might want to know which optional components (leather seats or alloy wheels, for example) are the most popular. Using Internet of Things technology,it is possible to:

Use sensors to detect which areas in a showroom are the most popular, and where customers linger longest;

Drill down into the available sales data to identify which components are selling fastest;

Automatically align sales data with supply, so that popular items don't go out of stock.

The information picked up by connected devices enables to make smart decisions about which components to stock up on, based on real-time information, which helps save time and money.

With the insight provided by advanced analytics comes the power to make processes more efficient. Smart objects and systems mean you can automate certain tasks, particularly when these are repetitive, mundane, time-consuming or even dangerous.

DESIGN PRINCIPLES AND NEEDED CAPABILITIES

In the near future, our everyday lives will be more and more filled with intelligent, connected objects. They will appear in our homes, in our working environments and in the cities, we live in as well as travel with us everywhere we go in the form of wearable, smart clothing and things we cannot even imagine right now. This development is called the internet of things, IoT.

IoT solutions consist of multiple elements: physical devices like sensors, actuators and interactive devices, the network connecting these devices, the data gathered from these devices and analyzed to create a meaningful experience and last but definitely not least, the physical context in which user interacts with the solution.

Design principles

1. Do your research

When designing IoT-enabled products, designers might make the mistake of forgetting why customers value these products in the first place. That's why it's a good idea to think about the value an IoT offering should deliver at the initial phase of your design.

When getting into IoT design, you're not building products anymore. You're building services and experiences that improve people's lives. That's why in-depth qualitative research is the key to figuring out how you can do that.

Assume the perspective of your customers to understand what they need and how your IoT implementation can solve their pain points. Research your target audience deeply to see what their existing experiences are and what they wish was different about them.

2. Concentrate on value

Early adopters are eager to try out new technologies. But the rest of your customer base might be reluctant to put a new solution to use. They may not feel confident with it and are likely to be cautious about using it.

If you want your IoT solution to become widely adopted, you need to focus on the actual tangible value it's going to deliver to your target audience.

What is the real end-user value of your solution? What might be the barriers to adopting new technology? How can your solution address them specifically?

Note that the features the early tech adopters might find valuable might turn out to be completely uninteresting for the majority of users. That's why you need to carefully plan which features to include and in what order, always concentrating on the actual value they provide.

3. Don't forget about the bigger picture

One characteristic trait of IoT solutions is that they typically include multiple devices that come with different capabilities and consist of both digital and physical touchpoints. Your solution might also be delivered to users in cooperation with service providers.

That's why it's not enough to design a single touchpoint well. Instead, you need to take the bigger picture into account and treat your IoT system holistically.

Delineate the role of every device and service. Develop a conceptual model of how users will perceive and understand the system. All the parts of your system need to work seamlessly together. Only then you'll be able to create a meaningful experience for your end-users.

4. Remember about the security

Don't forget that IoT solutions aren't purely digital. They're located in the real-world context, and the consequences of their actions might be serious if something goes wrong. At the same time, building trust in IoT solutions should be one of your main design drivers.

Make sure that every interaction with your product builds consumer trust rather than breaking it. In practice, it means that you should understand all the possible error situations that may be related to the context of its use. Then try to design your product in a way to prevent them. If error situations occur, make sure that the user is informed appropriately and provided with help.

Also, consider data security and privacy as a key aspect of your implementation. Users need to feel that their data is safe, and objects located in their workspaces or home can't be hacked. That's why quality assurance and testing the system in the real-world context are so important.

5. Build with the context in mind

And speaking of context, it pays to remember that IoT solutions are located at the intersection of the physical and digital world. The commands you give through digital interfaces produce real-world effects. Unlike digital commands, these actions may not be easily undone.

In a real-world context, many unexpected things may happen. That's why you need to make sure that the design of your solution enables users to feel safe and in control at all times.

The context itself is a crucial consideration during IoT design. Depending on the physical context of your solution, you might have different goals in mind. For example, you might want to minimize user distraction or design devices that will be resistant to the changing weather conditions.

The social context is an important factor, as well. Don't forget that the devices you design for workspaces or homes will be used by multiple users.

Make good use of prototypes

IoT solutions are often difficult to upgrade. Once the user places the connected object somewhere, it might be hard to replace it with a new version – especially if the user would have to pay for the upgrade.

Even the software within the object might be hard to update because of security and privacy reasons. Make sure that your design practices help to avoid costly hardware iterations. Get your solution right from the start. From the design perspective, it means that prototyping and rapid iteration will become critical in the early stages of the project.

Needed capabilities

1. Connectivity

It starts with how a device or sensor connects to the internet and a cloud platform. There are many options to choose from Wi-Fi through a hub or gateway, 2G, or 3G cellular networks. Once you have connectivity in place, now you can get the device or sensor talking to your cloud IoT platform. Ensure you find a service provider that can send data through clean API's that are easy to implement and install. This will ensure you can get quickly setup and start capturing your data within minutes.

2. Control

The next capability necessary when evaluating an IoT data platform is control of the device. There are a number of different scenarios for control including controlling a device through an application, device-to-device communication, or control from the cloud (based on an event, rule or some other pre-determined condition). For example, if you have a water leak detector, it can automatically send a command to the device which could be an appliance or part of the core infrastructure to turn off the water valve. Here, using two-way communication, a signal can be sent from the detector to the device via the cloud to shut off the water. Lastly, you can program the device from an app (or website) to shut off at a certain time or schedule based on a pre-programmed rule.

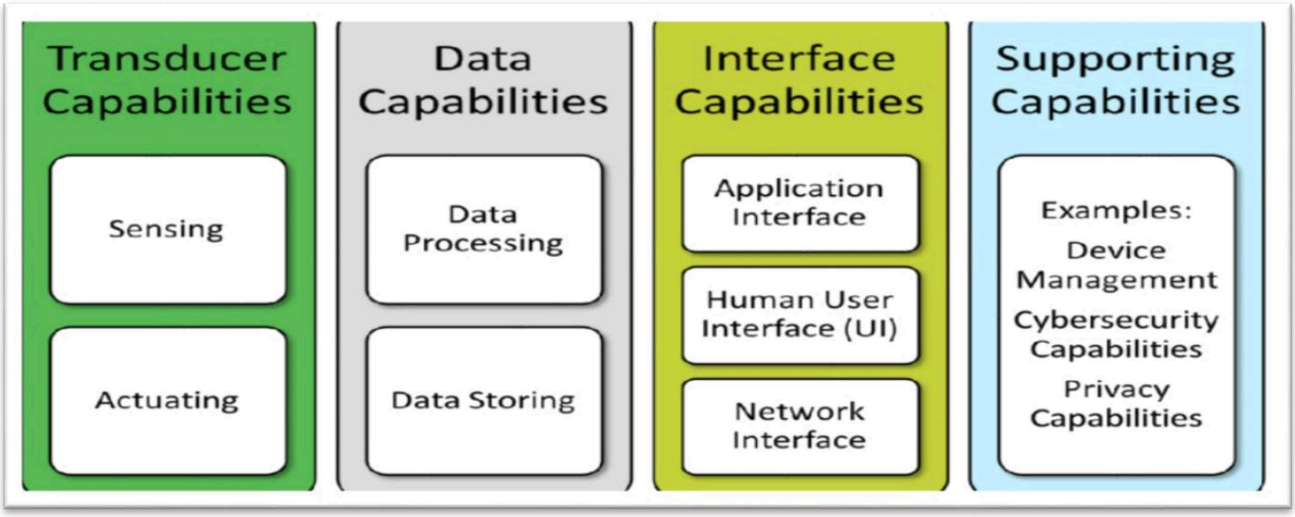
3. Device Management

Device management is also a major consideration. To keep devices and sensors up to date and functional, a strong device management solution is a core component of an IoT cloud platform. There are a few main capabilities a device management platform provides, including the ability for manufacturers to send software or firmware updates OTA (over-the-air), factory provisioning, as well as an out-of-box experience (OOBE).OOBE is part of a core experience that is often left to the last minute or completely glossed over. It's the first experience that an end user, be it a consumer, installer or technician has when interacting with a device for the first time.

4. Actionable Data

The last capability you should consider in a IoT data platform is how you can query the data in a manner that is clear and meaningful. It's one thing to get all your data in place, but the value of the data is only realized when it's turned into information that can help solve a problem. We want organizations to focus on t

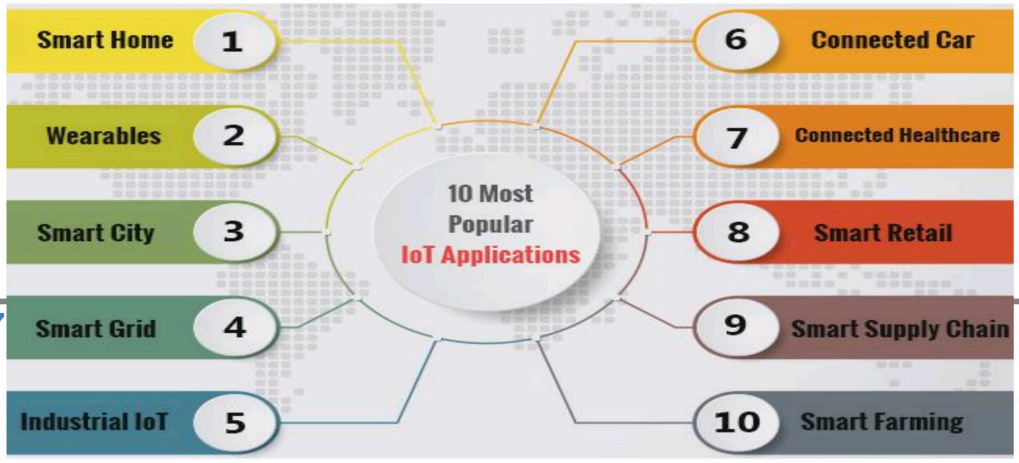
heir core competency, like making great appliances or services that deliver value to their customers, rather than focusing on cloud infrastructure that makes it possible.



IOT APPLICATIONS, SENSING AND ACTUATION

IoT Applications

The IoT applications are addressing the societal needs and the advancements to enabling technologies such as nano-electronics and cyber-physical systems continue to be challenged by a variety of technical (i.e., scientific and engineering), institutional, and economical issues.



List of IOT Applications

Smart Home
Wearable
Smart City
Smart Grid
Industrial IoT
Connected Car
Connected Healthcare
Smart Retail
Smart Transportation and Mobility
Smart Agriculture/ Farming

Smart Home

Smart Home has become the revolutionary ladder of success in the residential spaces and it is predicted Smart homes will become as common as smartphones. The cost of owning a house is the biggest expense in a homeowner's life. Smart Home products are promised to save time, energy and money.

Wearable

Wearable devices are installed with sensors and software which collect data and information about the users. This data is later pre-processed to extract essential insights about user. These devices broadly cover fitness, health and entertainment requirements. The pre-requisite from internet of things technology for wearable applications is to be highly energy efficient or ultra-low power and small sized.

Smart City

Smart cities use IoT devices such as connected sensors, lights, and meters to collect and analyze data. The cities then use this data to improve infrastructure, public utilities and services, and more.

Smart city devices work to make everyday tasks easier and more efficient, while relieving pain points related to public safety, traffic, and environmental issues.

Smart Grid

The basic idea behind the smart grids is to collect data in an automated fashion and analyze the behavior or electricity consumers and suppliers for improving efficiency as well as economics of electricity use. Smart Grids will also be able to detect sources of power outages more quickly and at individual household levels like nearby solar panel, making possible distributed energy system.

Industrial IoT

Industrial internet of things is empowering industrial engineering with sensors, software and big data analytics to create brilliant machines. IoT holds great potential for quality control and sustainability. Applications for tracking goods, real time information exchange about inventory among suppliers and retailers and automated delivery will increase the supply chain efficiency.

Connected Cars

A connected car is a vehicle which is able to optimiseits own operation, maintenance as well as comfort of passengers using on-board sensors and internet connectivity.

Most large auto makers as well as some brave startups are working on connected car solutions. Major brands like Tesla, BMW, Apple, Google are working on bringing the next revolution in automobiles.

Connected Healthcare

It can be used for out-patient care by healthcare providers, letting them get ECG, heart rate, respiratory rate, skin temperature, body posture, fall detection, and activity readings remotely. This can alert doctors to potential health problems before they arise, or give them additional insights into which treatments will be most effective for their patients, even when their patients aren't in the office.

Smart Retail

Today, retail stores are constantly focusing on leveraging the emerging technologies like cloud, mobile, RFID, beacons, etc., to provide connected retail services and better shopping experience to customers. For example, store owners are integrating sensors in the key zones of retail stores and connecting them to cloud through a gateway that enables real-time data analysis related to products, sales, and customers from these sensors.

Interestingly, IoT in retail and connected technologies are taking the retail industry by storm. 96% retailers are ready to make changes required to implement the Internet of Things in their stores.

Smart Transportation and Mobility

Internet of Vehicles (IoV) connected with the concept of Internet of Energy (IoE) represent future trends for smart transportation.

IoT technology that includes vehicle monitoring and maintenance, real-time tracking of packages, environmental sensors in shipping containers, information-gathering on employees and tools, and a number of safety-enhancing features for vehicles and people.

Smart Agriculture/ Farming

Farmers are using meaningful insights from the data to yield better return on investment. Sensing for soil moisture and nutrients, controlling water usage for plant growth and determining custom fertiliser are some simple uses of IoT.

Sensors and Actuators

Sensors and actuators have a number of similarities and dissimilarities in the functioning or processing. Here we have listed the differences between the actuator and a sensor.

The main difference between an actuator and a sensor is that the sensor converts the phyical gesture into electrical signals and do different works. Whereas, the actuator is responsible for the conversion of electrical signal to mechanical work.

Sensors measure discrete as well as continuous process variables. On the other hand, actuators are used to impel the parameters of both discrete and continuous processes.

Sensors are widely used to original electrical signals in different electrical application. On the other hand, actuators are very useful in the production of energy in the form of heat and motion.

Sensors are used as an input device because of the reason it is placed at input dork of the machine. However, actuator are used as output device as it is mostly placed at output port of the machinery.

Sensors are the one which acts as a brain because it provides information to do work.

Difference between Sensor and Actuator:

SENSOR	ACTUATOR
It converts physical characteristics into electrical signals.	It converts electrical signals into physical characteristics.
It takes input from environment.	It takes input from output conditioning unit of system.
It gives output to input conditioning unit of system.	It gives output to environment.
Sensor generated electrical signals.	Actuator generates heat or motion.
It is placed at input port of the system.	It is placed at output port of the system.
It is used to measure the physical quantity.	It is used to measure the continuous and discrete process parameters.
It gives information to the system about environment.	It accepts command to perform a function.
Example: Photo-voltaic cell which converts light energy into electrical energy.	Example: Stepper motor where electrical energy drives the motor.

Sensor

Sensor is a device used for the conversion of physical events or characteristics into the electrical signals. This is a hardware device that takes the input from environment and gives to the system by converting it.

For example, a thermometer takes the temperature as physical characteristic and then converts it into electrical signals for the system.

Actuator

Actuator is a device that converts the electrical signals into the physical events or characteristics. It takes the input from the system and gives output to the environment.

For example, motors and heaters are some of the commonly used actuators.

In a typical **IoT** system, a **sensor** may collect information and route to a control center. There, previously defined logic dictates the decision. As a result, a corresponding command controls an **actuator** in response to that sensed input. Thus, **sensors and actuators in IoT** work together from opposite ends.

Types of sensors	Types of actuators
Humidity Sensors	Linear Actuators

Pressure Sensors	Rotary Actuators
Proximity Sensors	Hydraulic Actuators
Level Sensors	Pneumatic Actuators
Accelerometers	Electric Actuators
Gas Sensors	Thermal and Magnetic Actuators
Temperature sensor	Mechanical Actuators

BASICS OF NETWORKING, M2M AND IOTTECHNOLOGY

Basics of Networking

Today computer networks are everywhere. You will find them in homes, offices, factories, hospitals leisure centers etc.

Home and Office Networks

The network you have at home uses the same networking technologies, protocols and services that are used in large corporate networks and on the Internet.

The only real difference between an home network and a large corporate network is the size.

A home network will have between 1 and 20 devices and a corporate network will have many thousands.

If you are completely new to networking then the basic course will introduce you to the basic networking protocols used in small home/office networks and on the Internet.

Setting Up and building a Home Network will introduce some basic networking component and show you how to build a home network and connect it to the Internet.

Networking Types

Networks can be **wired** or **wireless** with most networks being a mixture of both.

Wired vs Wireless Networks

Early (pre-2008) networks were predominately wired.

Today however most networks will use a mixture of wired and wireless network.

Wired networks use Ethernet as the data link protocol. This is unlikely to change with the IOT, as IOT devices will be predominantly wireless.

Wired Networks- Advantages and Disadvantages

Advantages:

Ethernet ports are found on almost all laptops/PCs and netbooks even on those 8 years old.

Wired networks are faster than Wireless. Data rates were periodically increased from the original 10 megabits per second, to 1gigabits per second. Most home networks use 10-100Mbps.

More secure than Wireless

Disadvantages:

Need to Use cable which can be unsightly, difficult to run and expensive.

Can't be used easily between buildings (planning etc.).

Note a new technology that uses mains cable overcomes many of these disadvantages. powerline networking is common on home/small office networks

Not supported on Mobile phones and tablets.

Wireless Networks – Advantages and Disadvantages

Wireless networks use **Wi-fi** as the data link protocol. However other wireless options are being developed for the IOT (Internet of things).

Advantages:

Generally easier to set up.

Can be used both on home and public networks

No cables required.

Can be used with mobile phones and tablets.

Disadvantages

Generally Slower than wired networks.

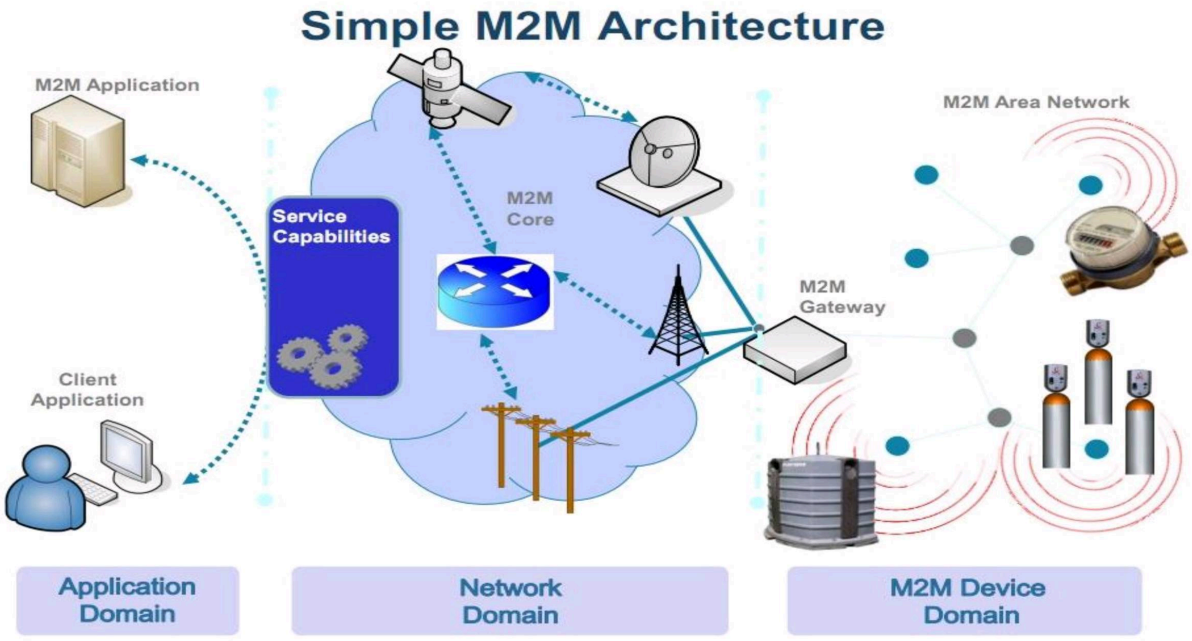
Limited by range.

Open to eavesdropping.

Not as secure depending on set up.

M2M

Shows the end-to-end architecture of M2M systems comprises of M2M area networks, communication networks and application domain.



Machine to machine (**M2M**) is direct communication between devices using any communications channel, including wired and wireless.

Machine to machine communication can include industrial instrumentation, enabling a sensor or meter to communicate the information it records (such as temperature, inventory level, etc.) to application software that can use it (for example, adjusting an industrial process based on temperature or placing orders to replenish inventory).

Such communication was originally accomplished by having a remote network of machines relay information back to a central hub for analysis, which would then be rerouted into a system like a personal computer.

More recent machine to machine communication has changed into a system of networks that transmits data to personal appliances. The expansion of IP networks around the world has made machine to machine communication quicker and easier while using less power. These networks also allow new business opportunities for consumers and suppliers.

M2M is not a new technology, primarily because it doesn't actually require any wireless technologies or complex digital devices. A wired connection between two machines can still be considered an M2M connection, and indeed the very first iterations of this technology used phone lines to communicate.

However, in the modern world, M2M usually refers to machines that communicate using things like WiFi or mobile networks. In this article, we're going to take a look at what technology underpins M2M, and how it's applied in the real world.

M2M generally centers around the concept of telemetry. This is simply where sensors of varying types will collect information, and then relay it to a central point of some description. In the past, this central point may have been a person or a personal computer.

The data would be interpreted and then used. This is still often the case today. However, modern M2M systems allow for far more complexity, allowing machines to communicate with one another and make decisions quickly.

IoT Technology

The Internet of Things (IoT) starts with connectivity, but since IoT is a widely diverse and multifaceted realm, you certainly cannot find a one-size-fits-all communication solution. Continuing our discussion on mesh and star topologies, in this article we'll walk through the six most common types of IoT wireless technologies.

1. LPWANs

Low Power Wide Area Networks (LPWANs) are the new phenomenon in IoT. By providing long-range communication on small, inexpensive batteries that last for years, this family of technologies is purpose-built to support large-scale IoT networks sprawling over vast industrial and commercial campuses.

LPWANs can literally connect all types of IoT sensors – facilitating numerous applications from **asset tracking, environmental monitoring** and **facility management** to **occupancy detection** and **consumables monitoring**. Nevertheless, LPWANs can only send small blocks of data at a low rate, and therefore are better suited for use cases that don't require high bandwidth and are not time-sensitive.

Also, not all LPWANs are created equal. Today, there exist technologies operating in both the licensed (NB-IoT, LTE-M) and unlicensed (e.g. MYTHINGS, LoRa, Sigfox etc.) spectrum with varying degrees of performance in key network factors. For example, while power consumption is a major issue for cellular-based, licensed LPWANs; Quality-of-Service and scalability are main considerations when adopting unlicensed technologies. Standardization is another important factor to think of if you want to ensure reliability, security, and interoperability in the long run.

2. Cellular (3G/4G/5G)

Well-established in the consumer mobile market, cellular networks offer reliable broadband communication supporting various voice calls and video streaming applications. On the downside, they impose very high operational costs and power requirements.

While cellular networks are not viable for the majority of IoT applications powered by battery-operated sensor networks, they fit well in specific use cases such as connected cars or fleet management in transportation and logistics. For example, in-car infotainment, traffic routing, advanced driver assistance systems (ADAS) alongside fleet telematics and tracking services can all rely on the ubiquitous and high bandwidth cellular connectivity.

Cellular next-gen 5G with high-speed mobility support and ultra-low latency is positioned to be the future of autonomous vehicles and augmented reality. 5G is also expected to enable real-time video surveillance for public safety, real-time mobile delivery of medical data sets for connected health, and several time-sensitive industrial automation applications in the future.

Also recommended for you: IoT Connectivity - 4 Latest Standards That Will Shape 2020 and beyond.

3. Zigbee and Other Mesh Protocols

Zigbee is a short-range, low-power, wireless standard (IEEE 802.15.4), commonly deployed in mesh topology to extend coverage by relaying sensor data over multiple sensor nodes. Compared to LPWAN, Zigbee provides higher data rates, but at the same time, much less power-efficiency due to mesh configuration.

Because of their physical short-range (< 100m), Zigbee and similar mesh protocols (e.g. Z-Wave, Thread etc.) are best-suited for medium-range IoT applications with an even distribution of nodes in close proximity. Typically, Zigbee is a perfect complement to Wi-Fi for various **home automation** use cases like smart lighting, HVAC controls, security and energy management, etc. – leveraging home sensor networks.

Until the emergence of LPWAN, mesh networks have also been implemented in industrial contexts, supporting several remote monitoring solutions. Nevertheless, they are far from ideal for many industrial facilities that are geographically dispersed, and their theoretical scalability is often inhibited by increasingly complex network setup and management.

4. Bluetooth and BLE

Defined in the category of Wireless Personal Area Networks, Bluetooth is a short-range communication technology well-positioned in the consumer marketplace. Bluetooth Classic was originally intended for point-to-point or point-to-multipoint (up to seven slave nodes) data exchange among consumer devices. Optimized for power consumption, Bluetooth Low-Energy was later introduced to address small-scale Consumer IoT applications.

BLE-enabled devices are mostly used in conjunction with electronic devices, typically smartphones that serve as a hub for transferring data to the cloud. Nowadays, BLE is widely integrated into fitness and medical wearables (e.g. smartwatches, glucose meters, pulse oximeters, etc.) as well as Smart Home devices (e.g. door locks) – whereby data is conveniently communicated to and visualized on smartphones.

The release of Bluetooth Mesh specification in 2017 aims to enable a more scalable deployment of BLE devices, particularly in retail contexts. Providing versatile indoor localization features, BLE beacon networks have been used to unlock new service innovations like in-store navigation, personalized promotions, and content delivery.

5. Wi-Fi

There is virtually no need to explain Wi-Fi, given its critical role in providing high-throughput data transfer for both enterprise and home environments. However, in the IoT space, its major limitations in coverage, scalability and power consumption make the technology much less prevalent.

Imposing high energy requirements, Wi-Fi is often not a feasible solution for large networks of battery-operated IoT sensors, especially in industrial IoT and smart building scenarios. Instead, it more pertains to connecting devices that can be conveniently connected to a power outlet like smart home gadgets and appliances, digital signages or security cameras.

6. RFID

Radio Frequency Identification (RFID) uses radio waves to transmit small amounts of data from an RFID tag to a reader within a very short distance. Till now, the technology has facilitated a major revolution in **retail** and **logistics**.

By attaching an RFID tag to all sorts of products and equipment, businesses can track their inventory and assets in real-time – allowing for better stock and production planning as well as optimized **supply chain management**. Alongside increasing IoT adoption, RFID continues to be entrenched in the retail sector, enabling new IoT applications like smart shelves, self-checkout, and smart mirrors.

What Technologies Have Made IoT Possible?

While the idea of IoT has been in existence for a long time, a collection of recent advances in a number of different technologies has made it practical.

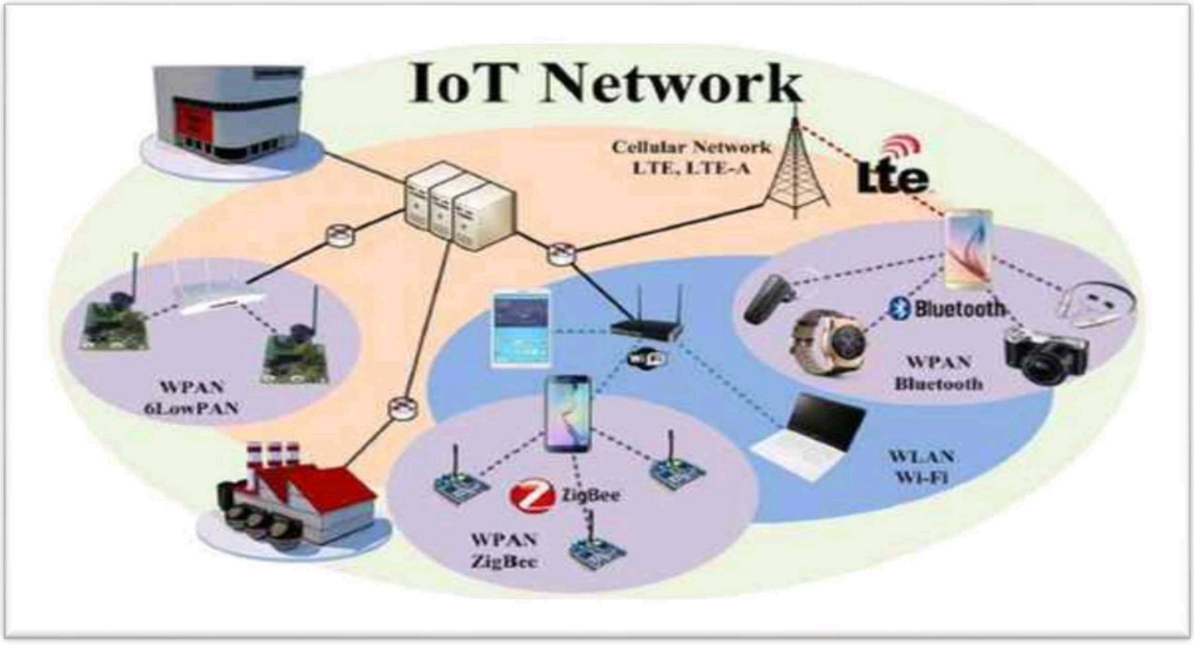
Access to low-cost, low-power sensor technology. Affordable and reliable sensors are making IoT technology possible for more manufacturers.

Connectivity. A host of network protocols for the internet has made it easy to connect sensors to the cloud and to other "things" for efficient data transfer.

Cloud computing platforms. The increase in the availability of cloud platforms enables both businesses and consumers to access the infrastructure they need to scale up without actually having to manage it all.

Machine learning and analytics. With advances in machine learning and analytics, along with access to varied and vast amounts of data stored in the cloud, businesses can gather insights faster and more easily. The emergence of these allied technologies continues to push the boundaries of IoT and the data produced by IoT also feeds these technologies.

Conversational artificial intelligence (AI). Advances in neural networks have brought natural-language processing (NLP) to IoT devices (such as digital personal assistants Alexa, Cortana, and Siri) and made them appealing, affordable, and viable for home use.



Differences between IoT and M2M

Communication Protocols:

Commonly uses M2M protocols include ZigBee, Bluetooth, ModBus, M-Bus, Wireless M-Bustec.

In IoT uses HTTP, CoAP, WebSocket, MQTT, XMPP, DDS, AMQP etc.,

Machines in M2M Vs Things inIoT:

Machines in M2M will be homogenous whereas Things in IoT will beheterogeneous.

Hardware Vs SoftwareEmphasis:

the emphasis of M2M is more on hardware with embedded modules, the emphasis of IoT is more on software.

Data Collection &Analysis

M2M data is collected in point solutions and often in on-premises storageinfrastructure.

The data in IoT is collected in the cloud (can be public, private or hybridcloud).

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- premisis enterpriseapplications.

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.

FUNDAMENTALS- DEVICES AND GATEWAYS

Devices

The sensing/actuating stage covers everything from legacy industrial devices to robotic camera systems, water level detectors, air quality sensors, accelerometers, and heart rate monitors. And the scope of the IoT is expanding rapidly, thanks in part to low-power wireless sensor network technologies and Power over Ethernet, which enable devices on a wired LAN to operate without the need for an A/C power source.

Physical devices and controllers that might control multiple devices. These are the thingsin the IoT, and they include a wide range of endpoint devices that send and receive information. Today, the list of devices is already extensive. It will become almost unlimited as more equipment is added to the IoT over time. Devices are diverse, and there are no rules about size, location, form factor, or origin. Some devices will be the size of a silicon chip. Some will be as large as vehicles. The IoT must support the entire range. Dozens or hundreds of equipment manufacturers will produce IoT devices.

Gateways

Simply put, an IoT gateway is a physical device or virtual platform that connects sensors, IoT modules, and smart devices to the cloud.

Gateways serve as a wireless access portal to give IoT devices access to the internet.

On the surface, it may sound like a simple router, enabling communication between different protocols and devices.

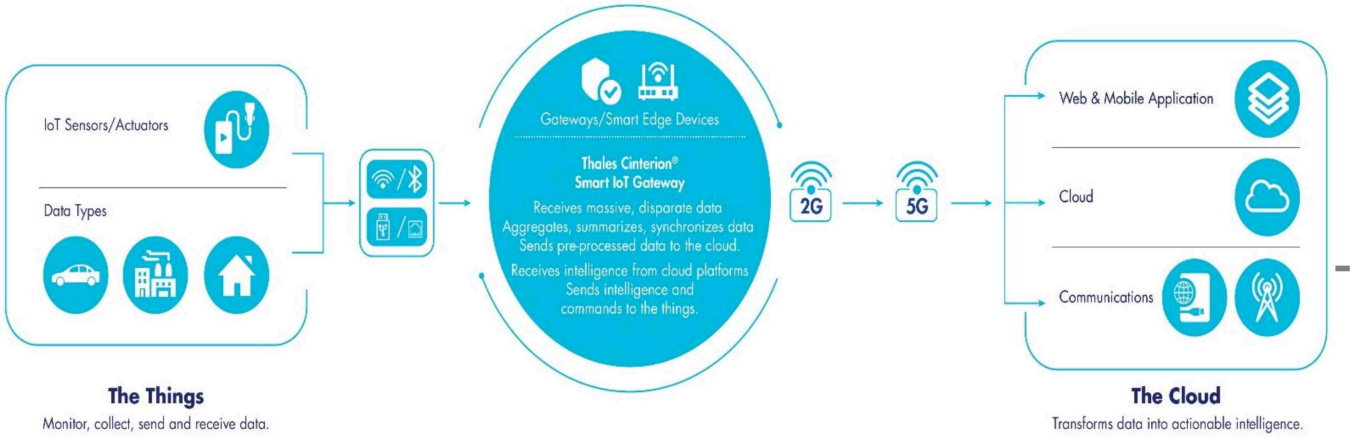
But IoT Gateways are sophisticated technology that does so much more, like edge-computing in particular.

An IoT Gateway collects massive data from many connected devices and sensors in any given IoT ecosystem.

The gateway pre-processes the data before passing it along to cloud platforms, where the heavy lifting of transforming data into meaningful intelligence is accomplished.

IoT gateways also receive information from the cloud, sent back to devices to allow autonomous management of devices in the field.

“This means that all the information moving through an IoT ecosystem – from an IoT device to



the cloud, or vice versa – goes through a connected IoT gateway."

DATA MANAGEMENT AND BUSINESS PROCESSES IN IOT

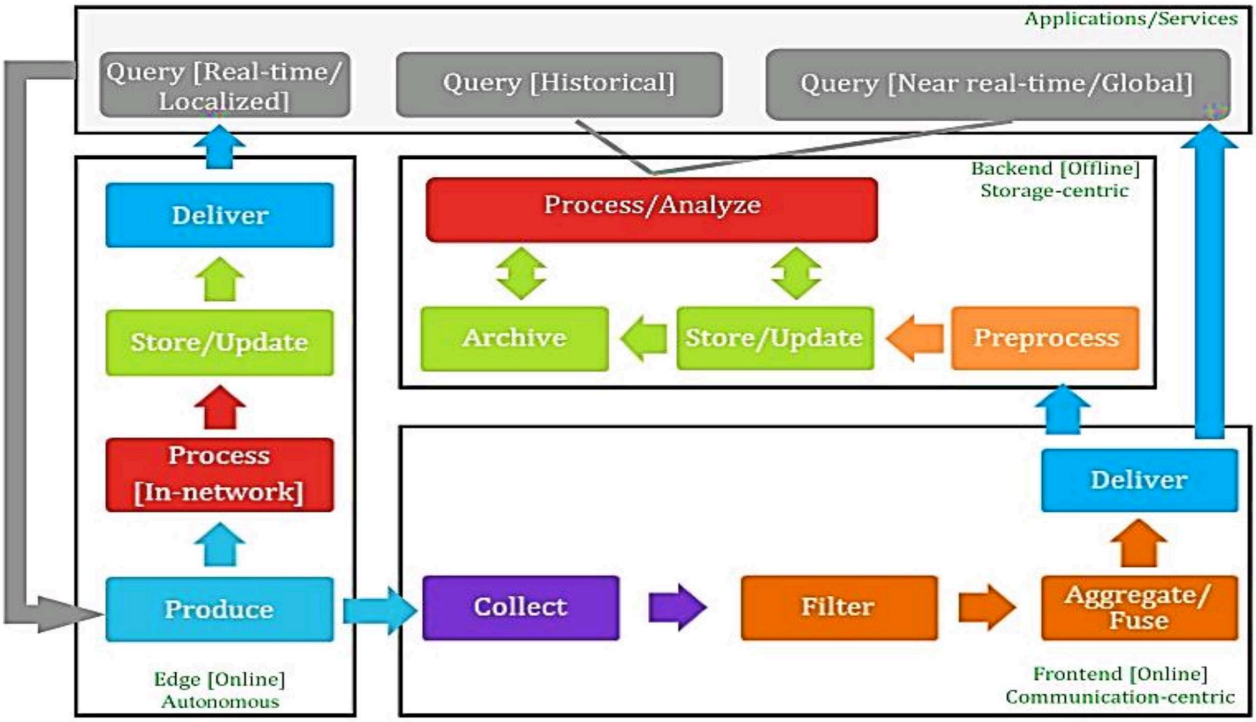
Data Management

Traditional data management systems handle the storage, retrieval, and update of elementary data items, records and files. In the context of IoT, data management systems must summarize data online while providing storage, logging, and auditing facilities for offline analysis. This expands the concept of data management from offline storage, query processing, and transaction management operations into online-offline communication/storage dual operations. We first define the data lifecycle within the context of IoT and then outline the energy consumption profile for each of the phases in order to have a better understanding of IoT data management.

IoT Data Lifecycle

The lifecycle of data within an IoT system—illustrated in Figure —proceeds from data production to aggregation, transfer, optional filtering and preprocessing, and finally to storage and archiving. Querying and analysis are the end points that initiate (request) and consume data production, but data production can be set to be "pushed" to the IoT consuming services. Production, collection, aggregation, filtering, and some basic querying and preliminary processing functionalities are considered online, communication-intensive operations. Intensive preprocessing, long-term storage and archival and in-depth processing/analysis are considered offline storage-intensive operations.

Storage operations aim at making data available on the long term for constant access/updates, while archival is concerned with read-only data. Since some IoT systems may generate, process, and store data in-network for real-time and localized services, with no need to propagate this data further up to concentration points in the system, "edges" that combine both processing and storage elements may exist as autonomous units in the cycle. In the following paragraphs, each of the elements in the IoT data lifecycle is explained



Querying: Data-intensive systems rely on querying as the core process to access and retrieve data. In the context of IoT, a query can be issued either to request real-time data to be collected for temporal monitoring purposes or to retrieve a certain view of the data stored within the system. The first case is typical when a (mostly localized) real-time request for data is needed. The second case represents more globalized views of data and in-depth analysis of trends and patterns.

Production: Data production involves sensing and transfer of data by the "Things" within the IoT framework and reporting this data to interested parties periodically (as in a subscribe/notify model), pushing it up the network to aggregation points and subsequently to database servers, or sending it as a response triggered by queries that request the data from sensors and smart objects. Data is usually time-stamped and possibly geo-stamped, and can be in the form of simple key-value pairs, or it may contain rich audio/image/video content, with varying degrees of complexity in-between.

Collection*:* The sensors and smart objects within the IoT may store the data for a certain time interval or report it to governing components. Data may be collected at concentration points or gateways within the network where it is further filtered and processed, and possibly fused into compact forms for efficient transmission. Wireless communication technologies such as Zigbee, Wi-Fi and cellular are used by objects to send data to collection points.

Aggregation*/Fusion*: Transmitting all the raw data out of the network in real-time is often prohibitively expensive given the increasing data streaming rates and the limited bandwidth. Aggregation and fusion techniques deploy summarization and merging operations in real-time to compress the volume of data to be stored and transmitted.

Delivery: As data is filtered, aggregated, and possibly processed either at the concentration points or at the autonomous virtual units within the IoT, the results of these processes may need to be sent further up the system, either as final responses, or for storage and in-depth analysis. Wired or wireless broadband communications may be used there to transfer data to permanent data stores.

Preprocessing: IoT data will come from different sources with varying formats and structures. Data may need to be preprocessed to handle missing data, remove redundancies and integrate data from different sources into a unified scheme before being committed to storage. This preprocessing is a known procedure in data mining called data cleaning.

***Storage**/Update—Archiving*: This phase handles the efficient storage and organization of data as well as the continuous update of data with new information as it becomes available. Archiving refers to the offline long-term storage of data that is not immediately needed for the system's ongoing operations. The core of centralized storage is the deployment of storage structures that adapt to the various data types and the frequency of data capture.

***Processing**/Analysis*: This phase involves the ongoing retrieval and analysis operations performed and stored and archived data in order to gain insights into historical data and predict future trends, or to detect abnormalities in the data that may trigger further investigation or action. Task-specific preprocessing may be needed to filter and clean data before meaningful operations take place.

Business Processes in IoT

The IoT is changing the way we live our lives and that is something that will only grow and grow, and it's certainly something that all businesses need to adapt to. There are some obvious benefits and some aspects that will require adjustments to processes. Here are some of the main changes and challenges facing companies as the IoT becomes more ever-present:

Data: As consumers use more and more devices that record data, there are opportunities for businesses to use this data for marketing and product development purposes, but only if the processes are in place to measure, analyze and report on this data. Business process management can automate this process and ensure that it remains effective and agile enough to keep pace with technological changes.

New ways of buying: The IoT gives consumers the chance to buy directly from their devices, whether it's an Amazon Echo or a smartphone or even that legendary refrigerator ordering fresh milk. Technology is making everything faster and more easily, so they will also be expecting faster deliveries and better service. BPM needs to be used to manage the processes that will allow this kind of development to meet the demand. IoT software and tools can help with this though, with inventories able to be tracked automatically.

Innovation: Whether it's new product development or upgrading existing products or services, the IoT offers the opportunities for businesses to deliver exciting new benefits for their customers.

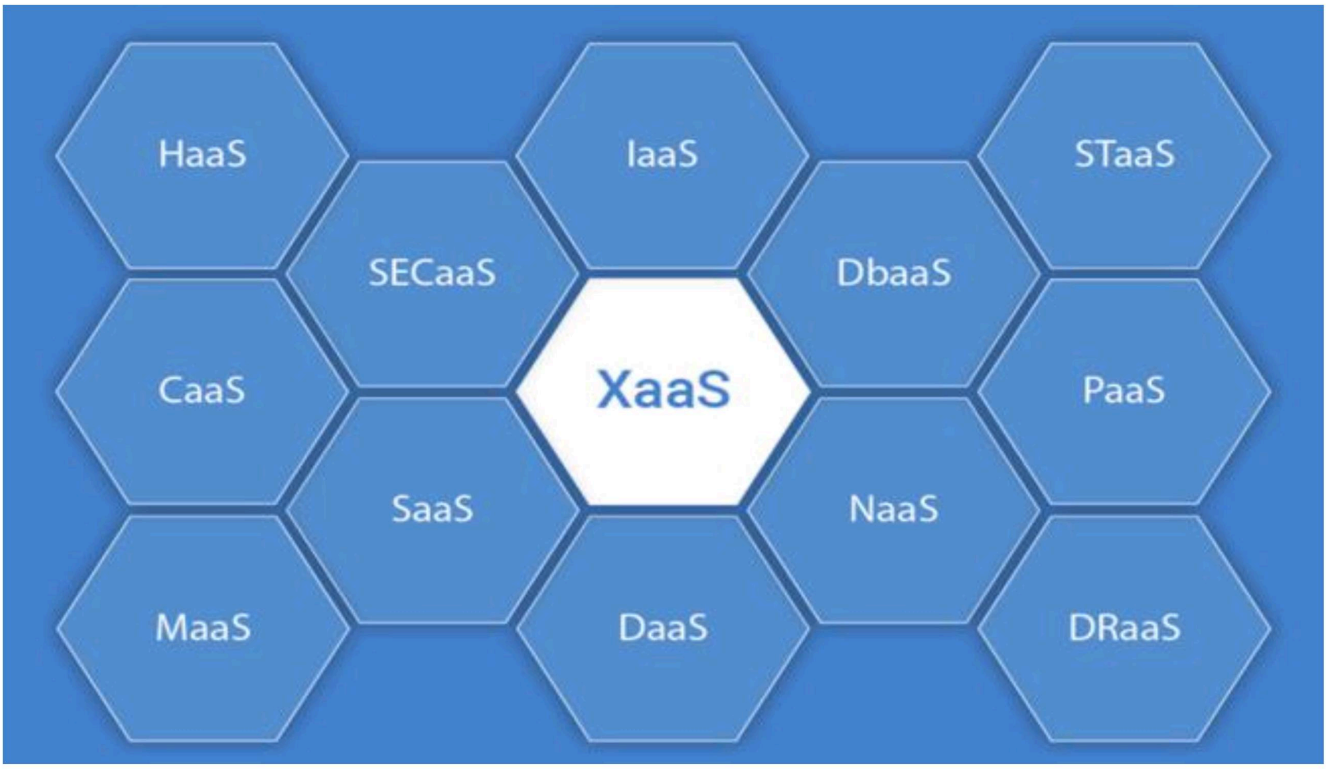
Customer service: Another area where processes need to be managed carefully because of the changes that the IoT have brought in is customer service. Products that utilize the internet should really be able to be fixed over the internet when something goes wrong. Consumers expect it and businesses should be able to deliver it, so BPM is needed to ensure that customer service processes are effective, efficient and easy to cope.

Centralized BPM: Business process management isn't simply something that is needed to make the IoT run more smoothly, the benefits can flow back in the opposite direction too. Integrating BPM software into devices means that the data can be analyzed from a central location and any changes can be fed back out again.

EVERYTHING AS A SERVICE (XAAS)

XaaS is a general, collective term that refers to the delivery of **anything as a service**. It recognizes the vast number of products, tools and technologies that vendors now deliver to users as a **service** over a network -- typically the internet -- rather than provide locally or on-site within an enterprise.

Most major companies now offer some form of XaaS, including Microsoft Azure, the various Amazon Web Services (AWS), and even Google Apps. In fact, if there are any types of IT services or computer-based functionality you require, then there is a high probability you can obtain it as XaaS. Some of the most common types of XaaS include:



Types of XaaS

Hardware-as-a-Service (HaaS)
Communication-as-a-Service (CaaS)
Monitoring as a Service (MaaS)
Security-as-a-Service (SECaaS)
Software-as-a-Service (SaaS)
Infrastructure as a service (IaaS)
Desktop-as-a-Service (DaaS)
Database-as-a-Service (DBaaS)
Network-as-a-service (NaaS)
Storage-as-a-Service (STaaS)
Platform-as-a service (PaaS)
Disaster recovery-as-a-service (DRaaS)

Hardware-as-a-Service (HaaS): Managed service providers (MSP) own some hardware and install it on customers' sites on demand. Customers utilize the hardware in accordance with service level agreements. This pay-as-you-go model is similar to leasing and can be compared to IaaS when computing resources are located at MSP's site and provided to users as virtual equivalents of physical hardware.

Communication-as-a-Service (CaaS): This model includes different communication solutions such as VoIP (voice over IP or Internet telephony), IM (instant messaging), video conference applications that are hosted in the vendor's cloud. A company can selectively deploy communication apps that best suit their current needs for a certain period and pay for this usage period only.

Monitoring-as-a-Service (MaaS) is a security service that provides security to IT assets of any business 24/7. It plays a vital role in securing an enterprise or government clients from any possible cyber threats. MaaS is a monitoring service that can be outsourced in a flexible and consumption-based subscription model.

Security-as-a-Service (SECaaS): This is the model of outsourced security management. A provider integrates their security services into your company's infrastructure and, as a rule, delivers them over the Internet. Such services may include anti-virus software, encryption, authentication, intrusion detection solutions and more.

Software-as-a-Service (SaaS): It is a cloud-based method of providing software to users. SaaS users subscribe to an application rather than purchasing it once and installing it. Users can log into and use a SaaS application from any compatible device over the Internet. The actual application runs in cloud servers that may be far removed from a user's location.

Infrastructure-as-a-Service(IaaS) is an instant computing infrastructure, provisioned and managed over the internet. A cloud computing service provider, such as Azure, manages the infrastructure, while you purchase, install, configure and manage your own software—operating systems, middleware and applications.

Desktop-as-a-Service (DaaS): Desktops are delivered as virtual services along with the apps needed for use. Thus, a client can work on a personal computer, using the computing capacities of third-party servers (which can be much more powerful than those of a PC).

Database-as-a-Service (DBaaS) is a cloud computing service model that provides users with some form of access to a database without the need for setting up physical hardware, installing software or configuring for performance. All of the administrative tasks and maintenance are taken care of by the service provider so that all the user or application owner needs to do is use the database. Of course, if the customer opts for more control over the database, this option is available and may vary depending on the provider.

Network-as-a-Service-(NaaS) brings Software Defined Networking (SDN), programmable networking and API-based operation to WAN services, transport, hybrid cloud, multi-cloud, Private Network Interconnect, and Internet Exchanges.Historic definitions focused on fundamental concepts of NaaS including: NaaS describes services for network transport connectivity.NaaS involves the optimization of resource allocations by considering network and computing resources as a unified whole.

Storage-as-a-Service (SaaS) is a cloud business model in which a company leases or rents its storage infrastructure to another company or individuals to store data.The client transfers the data meant for storage to the service provider on a set schedule over the SaaS provider's wide area network or over the Internet.

Platform-as-a-Service (PaaS) as the name suggests, provides you computing platforms which typically includes operating system, programming language execution environment, database, web server etc. Examples: AWS Elastic Beanstalk, Windows Azure, Heroku, Force.com, Google App Engine, Apache Stratos.

Disaster recovery-as-a-Service**(DRaaS)** is a cloud computing service model that allows an organization to back up its data and IT infrastructure in a third party cloud computing environment and provide all the DR orchestration, all through a SaaS solution, to regain access and functionality to IT infrastructure after a disaster. The as-a-service model means that the organization itself doesn't have to

own all the resources or handle all the management for disaster recovery, instead relying on the service provider.

ROLE OF CLOUD IN IOT AND SECURITY ASPECTS IN IOT

Role of Cloud in IoT

The Cloud is a centralized system that helps to deliver and transport data and various files across the Internet to data centers. The different data and programs 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 and analytics. The scalability of Cloud Computing means that as your business grows, your technological and analytical capabilities can too.

There are different types of Cloud services available, including Microsoft Azure Cloud development, and more information on each of these varying types of Cloud solutions can be found in our previous guide.

it is essential that both cloud and IoT form cloud-based IoT applications in a bid to make the most out of their combination. This alliance has led to the success of IoT. In addition to this, here are a few more pointers as to why the cloud is important from the point of view of IoT's success that are-

1.Provides remote processing power

Cloud as a technology empowers IoT to move beyond regular appliances such as air conditioners, refrigerators etc. This is because the cloud has such a vast storage that it takes away dependencies on on-premise infrastructure. With the rise of miniaturization and transition of 4G to higher internet speeds, the cloud will allow developers to offload fast computing processes.

2.Provides security and privacy

IoT's role in harnessing mobility is immense. However, its prowess would be incomplete without security. Cloud has made IoT more secure with preventive, detective and corrective controls. It has enabled users with strong security measures by providing effective authentication and encryption protocols. In addition to this, managing and securing the identity of users has been possible for IoT products with the help of biometrics. All of this is possible because of cloud's security.

3.Removes entry barrier for hosting providers

Today, many innovations in the field of IoT are looking at plug-and-play hosting services. Which is why the cloud is a perfect fit for IoT. Hosting **providers** do not have to depend on massive equipment or even any kind of hardware that will not support the agility IoT devices require. With the cloud, most hosting providers can allow their clients a ready-to-roll model, removing entry barriers for them.

4.Facilitates inter-device communication

Cloud acts as a bridge in the form of a mediator or communication facilitator when it comes to IoT. Many powerful APIs like Cloudflare, Cloud Cache and Drops are enabled by cloud communications, allowing easy linking to smartphones. This eases devices to talk to each other and not just us, which essentially is the tenet of IoT cloud.

It would be fair to say that cloud can accelerate the growth of IoT. However, deploying cloud technology also has certain challenges and shortcomings. Not because the cloud is flawed as a technology but the combination of IoT cloud can burden users with some obstacles. If you ever go ahead with an IoT cloud solution, it is better if you know the kind of challenges you may face in advance.

Security Aspect in IoT

As most of the systems are using existing wireless networks such aswifi, zigbee, zwave, GSM etc.IoT systems can be hacked using wireless devices. In order to have safe and secure use of IoT devices and IoT network, following precautions are advisable. These are very useful as IoT security aspects for both the user as well as IoT network service provider.

Do not store any critical business or personal data in internet cloud.

Do not store any password in your IoT device or anywhere in internet cloud.

Do not install any malware without verifying its authenticity.

Always install third-party software from authentic and genuine websites.

Do not be hurry in start using the IoT device, first secure your newly purchased IoT device with anti-malware and anti-virus softwares.

If possible, regularly change the password of IoT device in order to improve the security.

Do not bring any sensitive business material for re-work at home if home network is less secured compare to office network. Do not store such material in easily hackable storage devices or public storage locations. Moreover, avoid using wifi network for such work.

Switch off unused IoT devices as they are vulnerable for potential attack by hackers in a home network. For example, switch off IoT compliant thermostats when not needed.

Switch off wifi in your smartphone when you do not require internet access. This is because it has been found that smartphone-based fitness applications are vulnerable to leak passwords as well as location information easily over public wifi networks.

Business, finance and banking related companies should store the data and retain them till they are needed. Once they are no longer required, they should be deleted to minimize the possible hacking.

House owner should be cautious enough so that no unclaimed IoT device get installed or placed in their premises without their notice. As later these devices can be utilized by hackers for their bad intention.

IoT Service provider should provide regular software patches for smart watch, IoT sensors, IoT gadgets, healthcare applications used in smartphone etc. This helps IoT devices to be more secure. These patches should be robust enough to take care of modern and latest malwares and viruses.

Individually wireless networks based on various technologies are already been secured, which also helps avoid any possible security threats. Refer following links for further study on IoT security.

CHAPTER-2
ELEMENTS OF IOT

The IoT concepts imply a creation of network of various devices interacting with each other and with their environment. Interoperability and connectivity wouldn't be possible without hardware platforms that help developers solve issues such as building autonomous interactive objects or completing common infrastructure related tasks.

Let's go through the most popular IoT platforms and see how they work and benefit IoT software developers.

Arduino

1.The Arduino platform was created back in 2005 by the Arduino company and allows for open source prototyping and flexible software development and back-end deployment while providing significant ease of use to developers, even those with very little experience building IoT solutions.

2.The Arduino platform was created back in 2005 by the Arduino company and allows for open source prototyping and flexible software development and back-end deployment while providing significant ease of use to developers, even those with very little experience building IoT solutions.

3.Arduino is sensible to literally every environment by receiving source data from different external sensors and is capable to interact with other control elements over various devices, engines and drives. Arduino has a built-in micro controller that operates on the Arduino software.

4.Projects based on this platform can be both standalone and collaborative, i.e. realized with use of external tools and plug-in.

5.The integrated development environment (IDE) is composed of the open source code and works equally good with Mac, Linux and Windows OS. Based on a *processing* programming language, the Arduino platform seems to be created for new users and for experiments.

6.The processing language is dedicated to visualizing and building interactive apps using animation and Java Virtual Machine (JVM) platform.

7.Let's note that this programming language was developed for the purpose of learning basic computer programming in a visual context.

8. It is an absolutely free project available to every interested person. Normally, all the apps are programmed in C/C++, and are wrapped with *avr-gcc* (WinAVR in OS Windows).

9.Arduino offers analogue-to-digital input with a possibility of connecting light, temperature or sound sensor modules. Such sensors as *SPI* or *I2C* may also be used to cover up to 99% of these apps' market.

10.Arduino is a microcontroller (generally it is the 8-bit ATmega microcontroller), but not a mini-computer, which makes Arduino somehow limited in its features for advanced users. Arduino provides an excellent interactivity with external devices and offers a wide range of user manuals, project samples as well as a large community of users to learn from / share knowledge with.

Raspberry Pi	Arduino
Raspberry Pi is a Single Board Computer or SBC	Arduino is a Microcontroller based development board
It is based on Broadcom SoC, an ARM Cortex A Series Microprocessor	It is based on Atmel Microcontrollers. Arduino UNO uses ATmega328P Microcontroller
A Debian based Linux Distribution called Raspberry Pi OS is needed to boot the Raspberry Pi	As it is a Microcontroller, there is no need for an operating system
Raspberry Pi SBC can preform multiple tasks simultaneously due to its powerful processor and Linux based OS	Arduino is usually used for running a single task (or a very small no. of simple tasks) repeatedly, over and over again
All the necessary components like Processor, RAM, Storage, Connectors, GPIO Pins, etc. are situated on the Raspberry Pi Board itself	The Microcontroller on the Arduino Board (like ATmega328P) contains the Processor, RAM, ROM. The board contains supporting hardware (for power and data) and GPIO Pins
The cost of original Raspberry Pi SBC was $35. Subsequently, all the base variants of newer Raspberry Pi versions are priced at $35 only	The cost of original Arduino UNO is $23
Both the hardware and firmware of Raspberry Pi are closed-source i.e., it is not available for general use	Arduino is developed as open-source hardware and software from the beginning. You can easily get complete information on Arduino's hardware and software
Raspberry Pi SBC has several GPIO Pins (the famous 40-pin Raspberry Pi GPIO), using which you can connect different sensors, IO Devices, etc.	GPIO is an important peripheral of any Microcontroller and Arduino UNO is no exception. In Arduino terminology, these pins are called Digital IO (to connect LEDs and Buttons) and Analog IN (to connect analog devices)
Using the 40-pin GPIO Pins, you can add additional features / functionalities to Raspberry Pi with HAT (Hardware Attached on Top) expansion boards	A similar way to add extra features and functionalities in Arduino is using Arduino Shields (which are also connected through the IO Pins)
As Raspberry Pi is essentially a computer, you have to properly shutdown after using it or before powering it down	As Arduino is a Microcontroller board, you can plug and unplug the power as you want
The main programming languages for developing application in Raspberry Pi are Python, Scratch, Ruby, C, C++	Arduino can be programmed using C or C++ Programming Languages
The logic level of Raspberry Pi's GPIO is 3.3V. So, be careful when connecting hardware to the GPIO Pins	Arduino's logic level is 5V. As most of the sensors and modules are designed for Arduino, there won't be any problem connecting them to Arduino. But double check every module and connection just to be on the safe side
Raspberry Pi must be powered using an USB Power Adapter as it requires 5V 2A or 5V 3A power	Arduino can be powered from a computer's USB Port (make sure the USB Port's current limit is not exceeded)
You can easily connect to internet using Wi-Fi or Ethernet	For Arduino, you need additional module or shields to connect to internet
Raspberry Pi has the hardware for Bluetooth and Wi-Fi on board	There is no wireless connectivity in case of Arduino (at least on board)

Sensors and Actuators

Most IoT networks start from the object, or "thing," that needs to be connected.From an architectural standpoint, the variety of smart object types, shapes, and needs drive the variety of IoT protocols and architectures.

There are myriad ways to classify smart objects. One architectural classification could be: Battery-powered or power-connected:

This classification is based on whether the• object carries its own energy supply or receives continuous power from an external power source. Battery- powered things can be moved more easily than linepowered objects

. However, batteries limit the lifetime and amount of energy that the object is allowed to consume, thus driving transmission range and frequency

Mobile or static: This classification is based on whether the "thing" should move or• always stay at the same location. A sensor may be mobile because it is moved from one object to another (for example, a viscosity sensor moved from batch to batch in a chemical plant) or because it is attached to a moving object (for example, a location sensor on moving goods in a warehouse or factory floor). The frequency of the movement may also vary, from occasional to permanent. The range of mobility (from a few inches to miles away) often drives the possible power source.

Low or high reporting frequency: This classification is based on how often the• object should report monitored parameters. A rust sensor may report values once a month

. A motion sensor may report acceleration several hundred times per second. Higher frequencies drive higher energy consumption, which may create constraints on the possible power source (and therefore the object mobility) and the transmission range.

Simple or rich data: This classification is based on the quantity of data exchanged at• each report cycle.

A humidity sensor in a field may report a simple daily index value (on a binary scale from 0 to 255), while an engine sensor may report hundreds of parameters, from temperature to pressure, gas velocity, compression speed, carbon index, and many others. Richer data typically drives higher power consumption.

This classification is often combined with the previous to determine the object data throughput (low throughput to high throughput). You may want to keep in mind that throughput is a combined metric. A medium-throughput object may send simple data at rather high frequency (in which case the flow structure looks continuous), or may send rich data at rather low frequency (in which case the flow structure looks bursty).

Report range: This classification is based on the distance at which the gateway is• located. For example, for your fitness band to communicate with your phone, it needs to be located a few meters away at most.

The assumption is that your phone needs to be at visual distance for you to consult the reported data on the phone screen

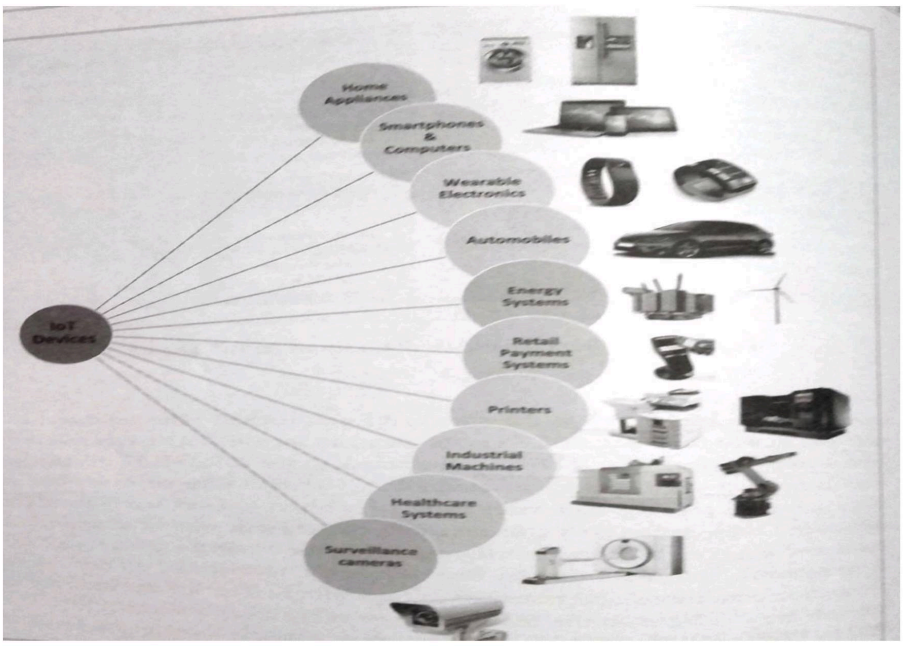
. If the phone is far away, you typically do not use it, and reporting data from the band to the phone is not necessary. By contrast, a moisture sensor in the asphalt of a road may need to communicate with its reader several hundred meters or even kilometers away.

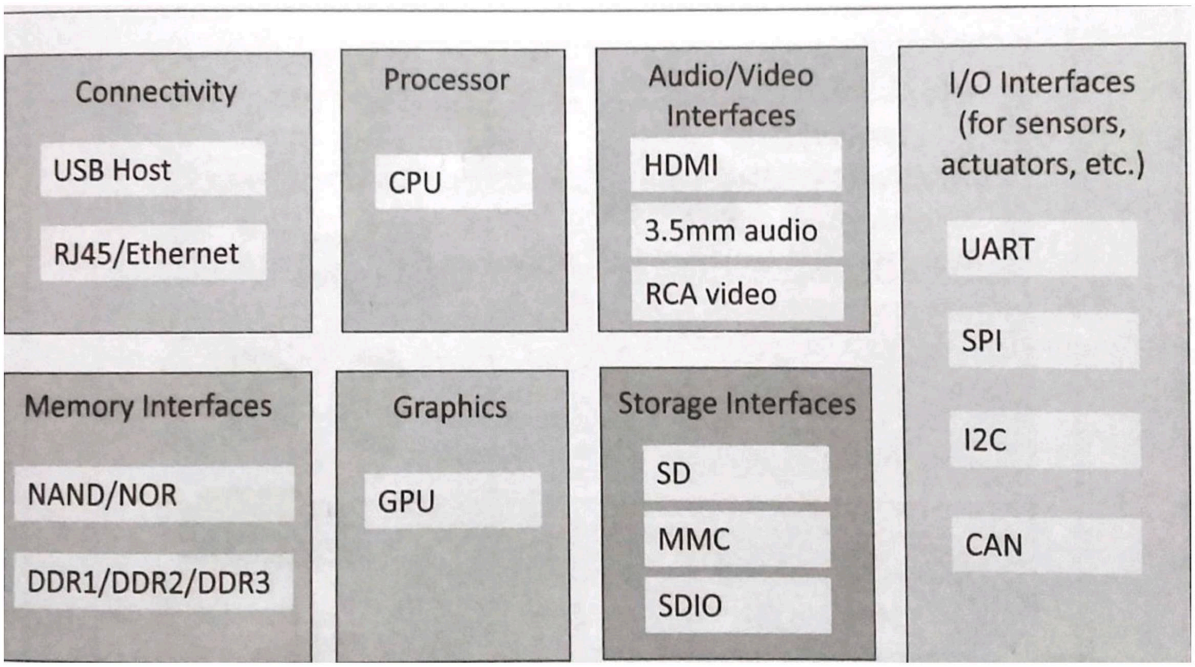
Object density per cell: This classification is based on the number of smart objects• (with a similar need to communicate) over a given area, connected to the same gateway.

An oil pipeline may utilize a single sensor at key locations every few miles. By contrast, telescopes like the SETI Colossus telescope at the Whipple Observatory deploy hundreds, and sometimes thousands, of mirrors over a small area, each with multiple gyroscopes, gravity, and vibration sensors.

Physical Design of IoT

Things in IoT:


IoT Devices
Home Appliances
Smartphones & Computers
Wearable Electronics
Automobiles
Energy Systems
Retail Payment Systems
Printers
Industrial Machines
Healthcare Systems
Surveillance cameras


Connectivity
USB Host
RJ45/Ethernet
Processor
CPU
Audio/Video Interfaces
HDMI
3.5mm audio
RCA video
I/O Interfaces (for sensors, actuators, etc.)
UART
SPI
I2C
CAN
Memory Interfaces
NAND/NOR
DDR1/DDR2/DDR3
Graphics
GPU
Storage Interfaces
SD
MMC
SDIO

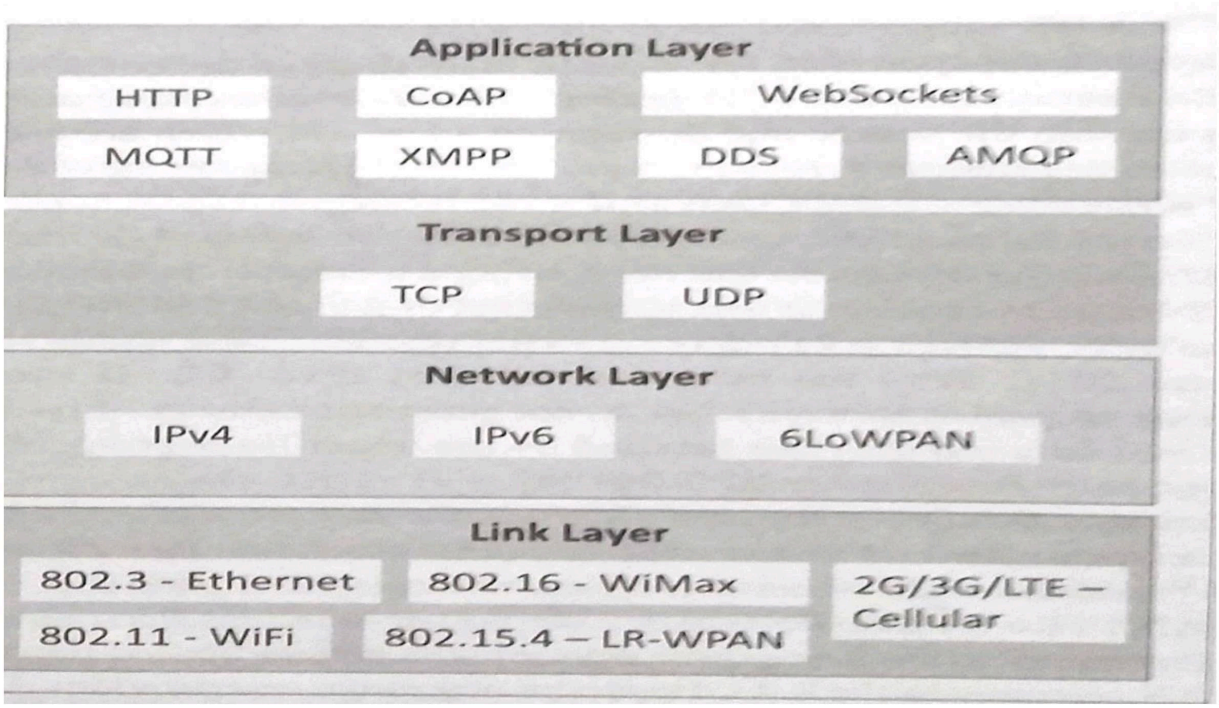
The things in IoT refers to IoT devices which have unique identities and perform remote sensing, actuating and monitoring capabilities. IoT devices can exchange data with other connected devices applications. It collects data from other devices and process data either locally or remotely.

An IoT device may consist of several interfaces for communication to other devices both wired and wireless. These includes (i) I/O interfaces for sensors, (ii) Interfaces for internet connectivity

memory and storage interfaces and (iv) audio/video interfaces.

IoT Protocols:

Link Layer : Protocols determine how data is physically sent over the network_s physical layer or medium. Local network connect to which host is attached. Hosts on the same link exchange data packets over the link layer using link layer protocols. Link layer determines how packets are coded and signaled by the h/w device over the medium to which the host is attached.



Protocols:

802.3-Ethernet: IEEE802.3 is collection of wired Ethernet standards for the link layer. Eg: 802.3 uses co-axial cable; 802.3i uses copper twisted pair connection; 802.3j uses fiber optic connection; 802.3ae uses Ethernet over fiber.

802.11-WiFi: IEEE802.11 is a collection of wireless LAN(WLAN) communication standards including extensive description of link layer. Eg: 802.11a operates in 5GHz band, 802.11b and 802.11g operates in 2.4GHz band, 802.11n operates in 2.4/5GHz band, 802.11ac operates in 5GHz band, 802.11ad operates in 60Ghzband.

802.16 - WiMax: IEEE802.16 is a collection of wireless broadband standards including exclusive description of link layer. WiMax provide data rates from 1.5 Mb/s to 1Gb/s.

802.15.4-LR-WPAN: IEEE802.15.4 is a collection of standards for low rate wireless personal area network(LR-WPAN). Basis for high level communication protocols such as ZigBee. Provides data rate from 40kb/s to250kb/s.

2G/3G/4G-Mobile Communication: Data rates from 9.6kb/s(2G) to up to100Mb/s(4G).

Network/Internet Layer: Responsible for sending IP datagrams from source n/w to destination n/w. Performs the host addressing and packet routing. Datagrams contains source and destination address.

Protocols:

IPv4: Internet Protocol version4 is used to identify the devices on a n/w using a hierarchical addressing scheme. 32 bit address. Allows total of 2**32addresses.

IPv6: Internet Protocol version6 uses 128 bit address scheme and allows 2**128 addresses.

6LOWPAN:(IPv6overLowpowerWirelessPersonalAreaNetwork)operates in 2.4 GHz frequency range and data transfer 250 kb/s.

Transport Layer: Provides end-to-end message transfer capability independent of the underlying n/w. Set up on connection with ACK as in TCP and without ACK as in UDP. Provides functions such as error control, segmentation, flow control and congestion control. **Protocols:**

TCP: Transmission Control Protocol used by web browsers(along with HTTP and HTTPS), email(along with SMTP, FTP). Connection oriented and stateless protocol. IP Protocol deals with sending packets, TCP ensures reliable transmission of protocols in order. Avoids n/w congestion and congestion collapse.

UDP: User Datagram Protocol is connectionless protocol. Useful in time sensitive applications, very small data units to exchange. Transaction oriented and stateless protocol. Does not provide guaranteed delivery.

Application Layer: Defines how the applications interface with lower layer protocols to send data over the n/w. Enables process-to-process communication using ports.

Protocols:

HTTP: Hyper Text Transfer Protocol that forms foundation of WWW. Follow request- response model Stateless protocol.

CoAP: Constrained Application Protocol for machine-to-machine (M2M) applications with constrained devices, constrained environment and constrained n/w. Uses client- server architecture.

WebSocket: allows full duplex communication over a single socket connection.

MQTT: Message Queue Telemetry Transport is light weight messaging protocol based on publish-subscribe model. Uses client server architecture. Well suited for constrained environment.

XMPP: Extensible Message and Presence Protocol for real time communication and streaming XML data between network entities. Support client-server and server-server communication.

DDS: Data Distribution Service is data centric middleware standards for device-to-device or machine-to-machine communication. Uses publish-subscribe model.

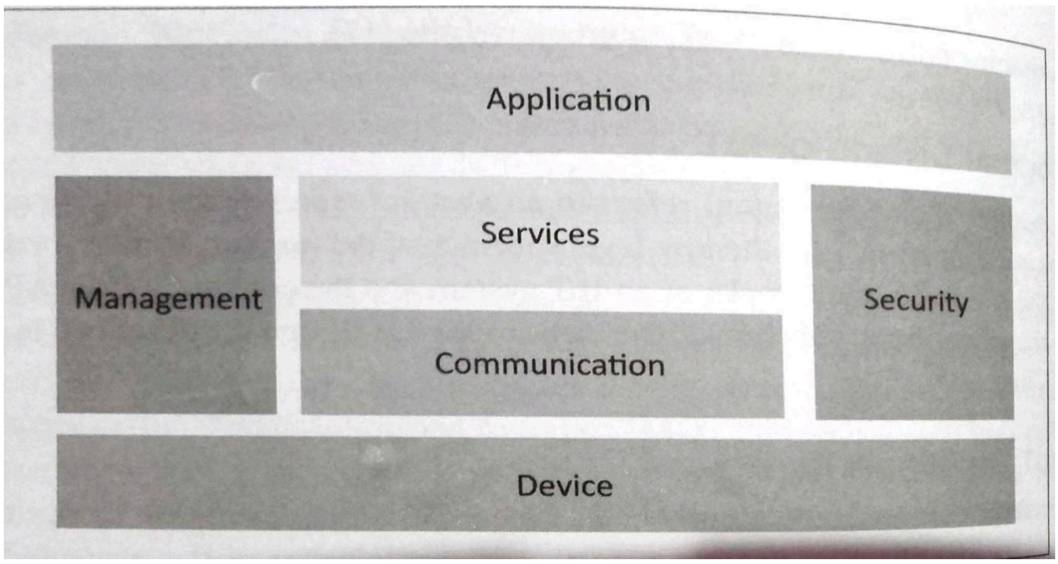
AMQP: Advanced Message Queuing Protocol is open application layer protocol for business messaging. Supports both point-to-point and publish-subscribe model.

LOGICAL DESIGN of IoT

Refers to an abstract represent of entities and processes without going into the low level specifies of implementation.

1) IoT Functional Blocks 2) IoT Communication Models 3) IoT Comm. APIs

IoT Functional Blocks: Provide the system the capabilities for identification, sensing, actuation, communication and management.



Device: An IoT system comprises of devices that provide sensing, actuation, monitoring and control functions.

Communication: handles the communication for IoTsystem.

Services: for device monitoring, device control services, data publishing services and services for device discovery.

Management: Provides various functions to govern the IoT system.

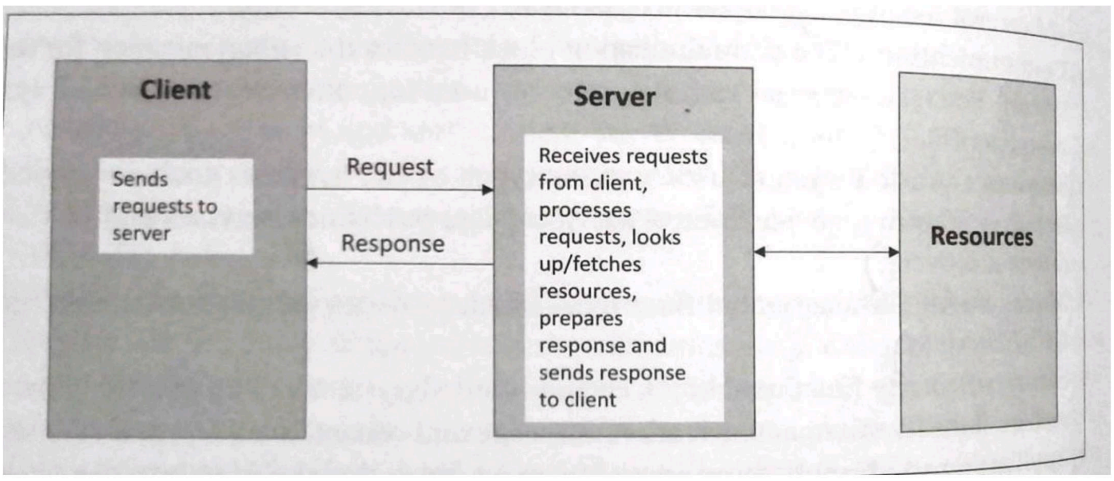
Security: Secures IoT system and priority functions such as authentication ,authorization, message and context integrity and data security.

Application: IoT application provide an interface that the users can use to control and monitor various aspects of IoT system.

IoT Communication Models:

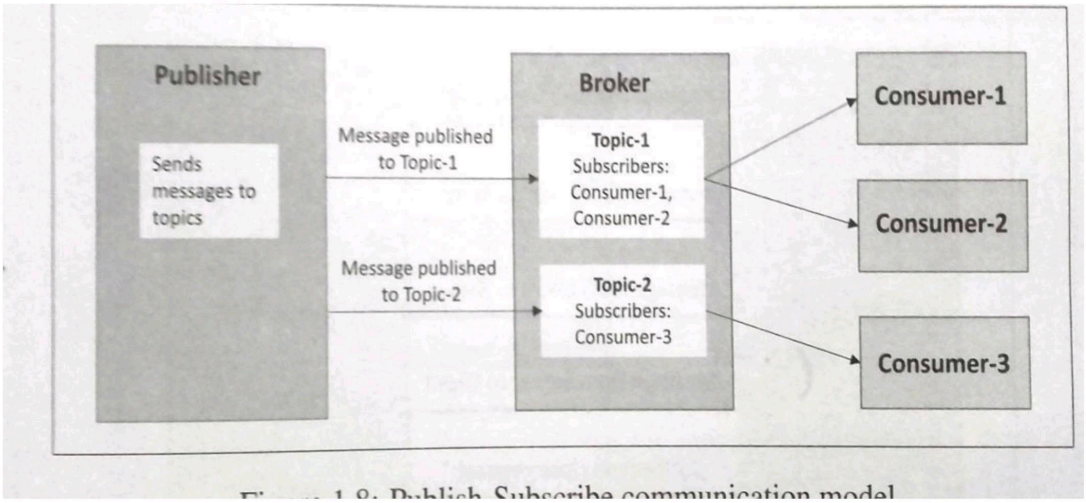
1) Request-Response 2) Publish-Subscibe 3)Push-Pull4) ExclusivePair

Request-Response Model:



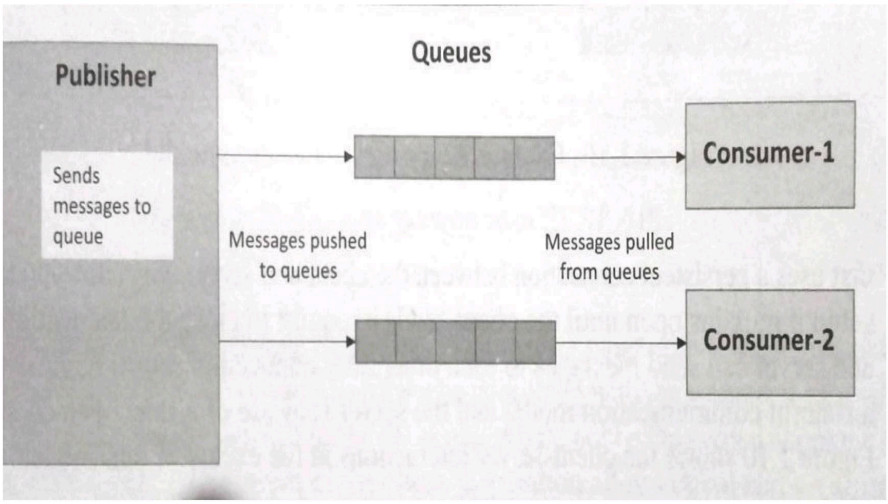
In which the client sends request to the server and the server replies to requests. Is a stateless communication model and each request-response pair is independent of others.

Publish-Subscibe Model:

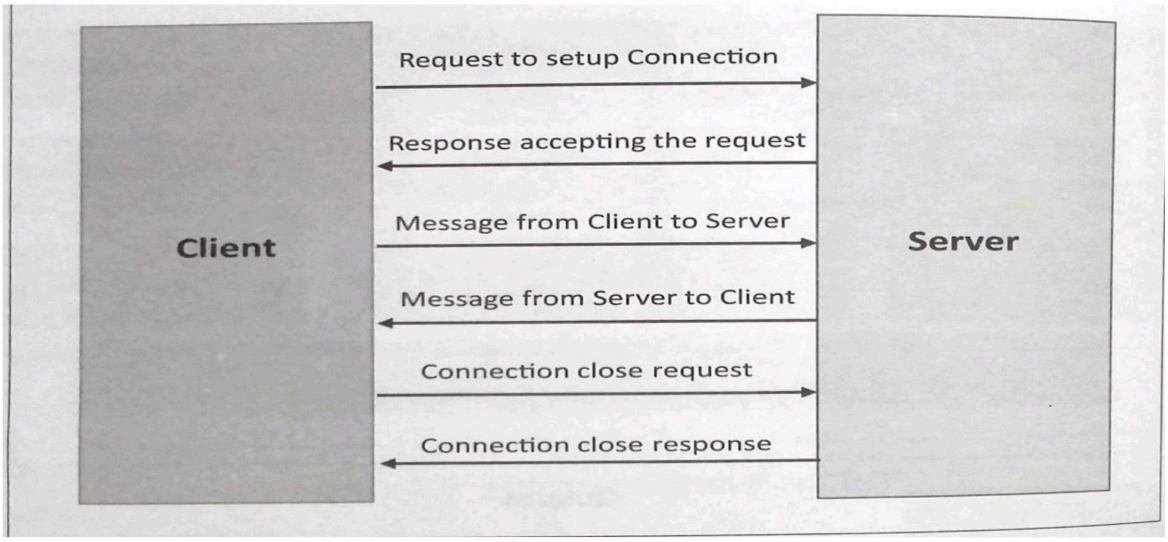


Involves publishers, brokers and consumers. Publishers are source of data. Publishers send data to the topics which are managed by the broker. Publishers are not aware of the consumers. Consumers subscribe to the topics which are managed by the broker. When the broker receives data for a topic from the publisher, it sends the data to all the subscribed consumers.

Push-Pull Model: in which data producers push data to queues and consumers pull data from the queues. Producers do not need to aware of the consumers. Queues help in decoupling the message between the producers and consumers.



Exclusive Pair: is bi-directional, fully duplex communication model that uses a persistent connection between the client and server. Once connection is set up it remains open until the client send a request to close the connection. Is a stateful communication model and server is aware of all the open



connections.

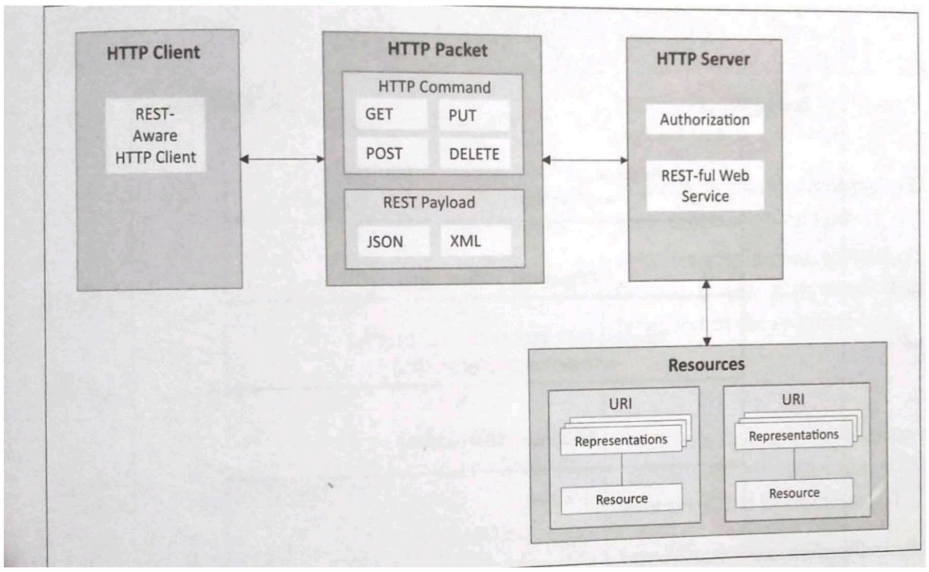
IoT Communication APIs:

REST based communication APIs(Request-Response Based Model)

WebSocket based Communication APIs(Exclusive PairBased Model)

REST based communication APIs: Representational State Transfer(REST) is a set of architectural principles by which we can design web services and web APIs that focus on a system_s resources and have resource states are addressed and transferred.

The REST architectural constraints: Fig. shows communication between client server with REST APIs.


HTTP Client
REST-
Aware
HTTP Client
HTTP Packet
HTTP Command
GET
PUT
POST
DELETE
REST Payload
JSON
XML
HTTP Server
Authorization
REST-ful Web
Service
Resources
URI
Representations
Resource
URI
Representations
Resource

Client-Server: The principle behind client-server constraint is the separation of concerns. Separation allows client and server to be independently developed and updated.

Stateless: Each request from client to server must contain all the info. Necessary to understand the request, and cannot take advantage of any stored context on the server.

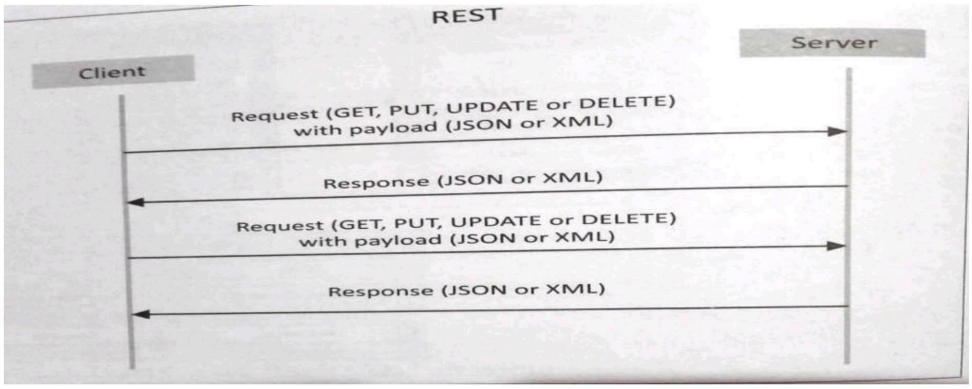
Cache-able: Cache constraint requires that the data within a response to a request be implicitly or explicitly labeled as cache-able or non-cacheable. If a response is cache-able, then a client cache is given the right to reuse that response data for later, equivalent requests.

Layered System: constraints the behavior of components such that each component cannot see beyond the immediate layer with which they are interacting.

User Interface: constraint requires that the method of communication between a client and a server must be uniform.

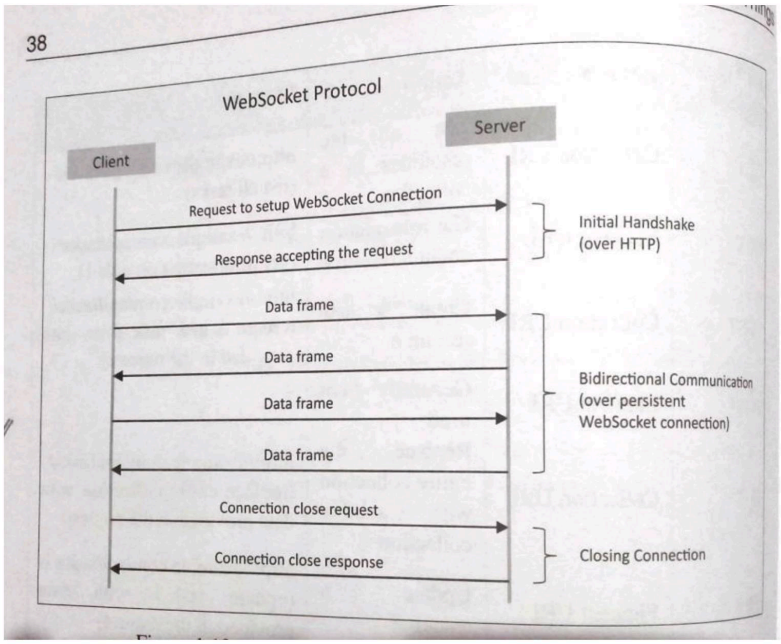
Code on Demand: Servers can provide executable code or scripts for clients to execute in their context. This constraint is the only one that is optional.

Request-Response model used by REST:



1.RESTful web service is a collection of resources which are represented by URIs. RESTful web API has a base URI(e.g: http://example.com/api/tasks/). The clients and requests to these URIs using the methods defined by the HTTP protocol(e.g: GET, PUT, POST or DELETE). A RESTful web service can support various internet media types.

2.WebSocket Based Communication APIs: WebSocket APIs allow bi-directional, full duplex communication between clients and servers. WebSocket APIs follow the exclusive pair communication model.



CHAPTER 3
IOT APPLICATION DEVELOPMENT

IoT Frameworks

For an IoT framework to be reliable and dependable, some minimal set of measures should be satisfied to achieve integration and interoperability in IoT. These frameworks span across the IoT research communities ranging from academic research to organisational research which focus on integrating things in IoT. Since IoT paradigm itself is still in evolving state, we propose a set of minimal measures to be satisfied by IoT frameworks for integration.

These are:

• Contract decoupling: An IoT system contains heterogeneous devices with disparate communication protocols. An integration framework should be competent enough to efficiently handle contract decoupling. Contract decoupling is the ability of service consumers and service producers to 4 IoT Architectural Framework independently evolve without terminating the contract between them For example, a service might be in a JSON format and the service consumer needs an input in XML. The framework should provide support to transform the message to the format that fulfils the contract between them.

• Scalability: Given the evolving nature of IoT and the predictions and calculations by and an efficient integration framework should be scalable and evolvable enough to support the billions of things soon to be connected to the internet.

• Ease of testing: An integration framework should support ease of testing and debugging. It should provide support for debugging defects and failures, integration testing, component testing, system testing, compatibility testing, installation test, functional and non-functional testing, performance testing and security testing.

• Ease of development: An IoT integration framework should provide a means of easy development for developers. The framework should exclude all complexities and provide proper documentation for non-developers and developers with basic programming knowledge to easily understand the internals of the framework.

• Fault tolerance: An IoT system has to be dependable and resilient. An intelligent integration framework should effectively handle faults as IoT devices can eventually toggle between offline and online states. The framework should provide self-healing mechanisms for transient faults (network faults, node level faults, etc.), unauthorised access error, server crash failure, omission failure (when the server does not receive incoming requests from client), timing fault, etc.

• Lightweight implementation: Integration frameworks should have a lightweight overhead both in its development and deployment stage. It should be lightweight and easy to install, uninstall, activate, deactivate, update, versioning and adaptable.

• Service coordination: Service coordination is the orchestration and choreography of services. Service orchestration is the coordination of multiple services by a mediator acting as a centralised component. Service choreography on the other hand, is the chaining of services together to execute a particular transaction. Integration frameworks should support at least either or both to achieve reliability.

• Inter domain operability: The framework should further be extensible to support inter domain communication. For example, in a smart car domain, an integration framework should also provide support for communication and interaction with traffic lights, road closure, etc. belonging to a smart city domain. Regardless of the research community or disparity in research, they all aim to achieve extensibility, flexibility, scalability, design reuse and implementation reuse. The next sub-sections will present an overview of some IoT frameworks.

IMPLEMENTATION OF DEVICE INTEGRATION IN IOT

There are five steps for integration leaders that can address some of the most common challenges in IoT integration. The five steps are:

Adopt an API-First Approach An API-first strategy is particularly relevant to IoT projects because they rely heavily on mobile and cloud computing, technologies which already use an API-centric approach. This should not, however, be misinterpreted as an API-only approach. APIs alone do not sufficiently address all the capabilities needed to securely and reliably scale up integration in large distributed systems.“Determine the integration requirements for a specific IoT solution that can’t be handled via APIs, and then assess whether the built-in integration capabilities of your IoT platform will suffice.

Identify Communication Requirements for IoT Devices First, identify how ‘things’ in your project will communicate, and select the best technology accordingly, ranging from cellular networks to short range wireless such as Bluetooth or ZigBee. It’s also important to consider factors such as the number, and type, of things and how different technologies will handle these variables. Then look for the network topology — considering the potential role for edge computing or gateways — that best suits the specific requirements for device autonomy, localized computing, device aggregation and so on. Once these areas have been qualified, it’s possible to assess whether a bundled IoT platform meets requirements for the project, or whether extra solutions will be needed to build a network for things.

Leverage Cloud for Data and Process Integration This step focuses on integrating IoT platforms with core business processes. Most Gartner clients report that the built-in integration capabilities of their IoT platform are good enough for initial deployments, such as an exploratory Mode 2 project.“Consider using your IoT platform for the initial implementation and then use a commercial integration solution, like an iPaaS platform, to scale up the project or support more complex integration, to implement workflow, or multiple IoT projects, or to access advanced integration features such as high-performance, general-purpose translation.

Selectively Use Traditional Software Most mid-to-large-sized organizations have substantial investments in traditional on-premises integration middleware. While these tools are generally not optimized for IoT device connectivity or cloud services integration, they can likely help if your project is on-premises, or if your IoT platform must integrate with data and applications that are mostly on-premises.

Use API Management Tools API management capabilities vary widely in different IoT platforms or other types of middleware. At minimum, you should plan for the possibility of adding a third-party API management solution to your IoT project, to ensure secure and reliable scaling as APIs proliferate. This is especially true if your project involves many APIs, or has APIs that connect with many consumers, are exposed on public networks, or return sensitive or restricted data.“IoT projects are often associated with disruptive business models and in the excitement, an appreciation of the potential complexity can be lost.

Raspberry Pi

1.Raspberry Pi is a mono-board computing platform that's as tiny as a credit card. Initially it was developed for computer science education with later on progress to wider functions.

2.Since the inception of Raspberry, the company sold out more than 8 million items. 3.Raspberry Pi 3 is the latest version and it is the first 64-bit computing board that also comes with built-in Wi-Fi and Bluetooth functions

4.According to Raspberry Pi Foundation CEO Eben Upton, "*it's been a year in the making*". The Pi3 version is replaced with a quad-core 64-bit 1.2 GHz ARM Cortex A53 chip, 1GB of RAM, VideoCore IV graphics, Bluetooth 4.1 and 802.11n Wi-Fi.

5.The developers claim the new architecture delivers an average 50% performance improvement over the Pi 2.

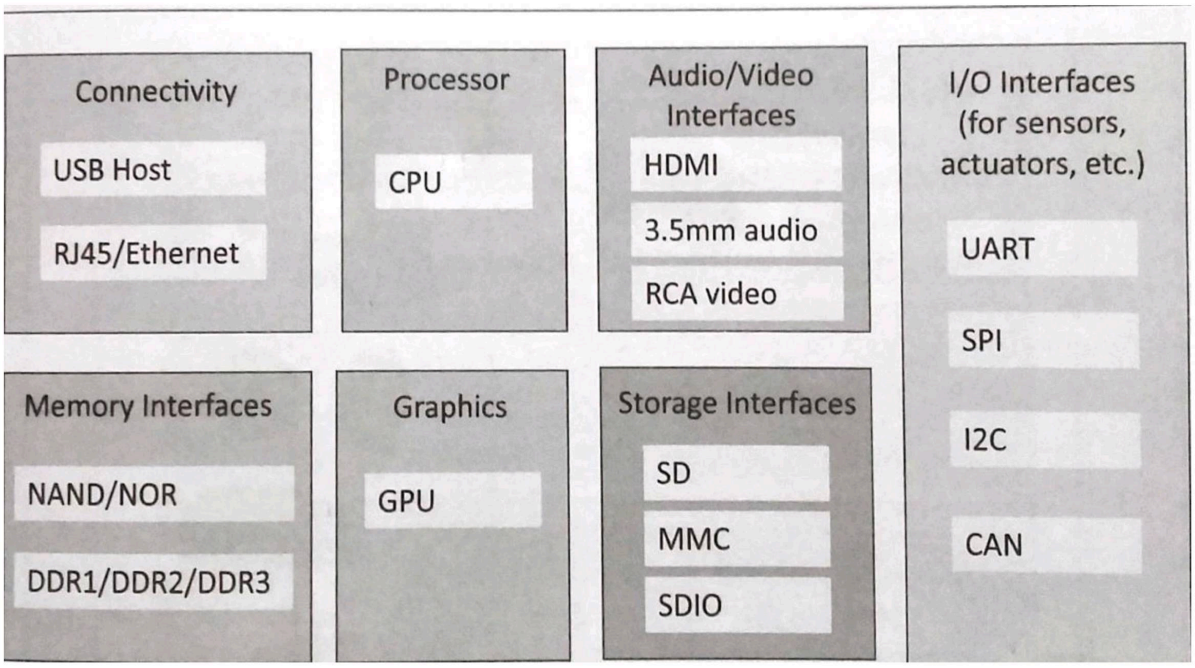
6.Another peculiarity of Raspberry Pi is the *GPIO* (General Purpose Input-Output), which is a low-level interface of self-operated control by input-output ports. Raspberry has it as a 40-pin connector.

7.Raspberry Pi uses Linux as its default operating system (OS). It’s also fully Android compatible. Using the system on Windows OS is enabled through any virtualization system like *XenDesktop.*

8.If you want to develop an application for Raspberry Pi on your computer, it is necessary to download a specific toolset comprised of ARM-compiler and some libraries complied down to ARM-target platform like *glibc.*

Physical Design of IoT

Things in IoT:


Connectivity
USB Host
RJ45/Ethernet
Processor
CPU
Audio/Video Interfaces
HDMI
3.5mm audio
RCA video
I/O Interfaces (for sensors, actuators, etc.)
UART
SPI
I2C
CAN
Memory Interfaces
NAND/NOR
DDR1/DDR2/DDR3
Graphics
GPU
Storage Interfaces
SD
MMC
SDIO

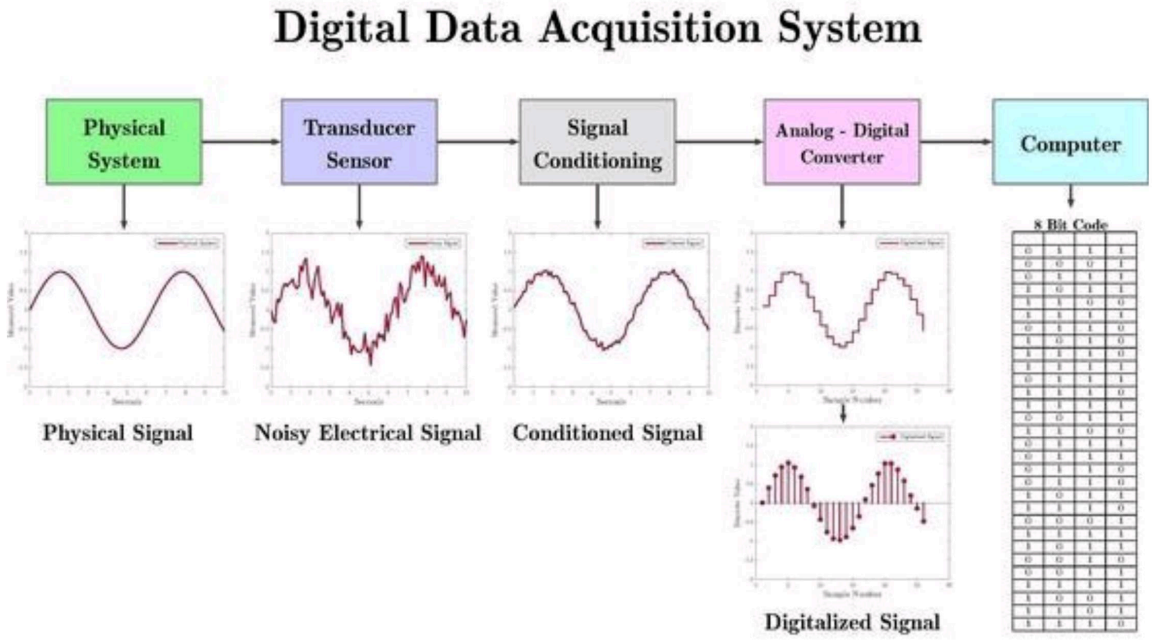
DAS

Data acquisition is the process of sampling signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer. Data acquisition systems, abbreviated by the initialisms *DAS, DAQ,* or *DAU,* typically convert analog waveforms into digital values for processing. The components of data acquisition systems include:

• Sensors, to convert physical parameters to electrical signals.

• Signal conditioning circuitry, to convert sensor signals into a form that can be converted to digital values.

• Analog-to-digital converters, to convert conditioned sensor signals to digital values



Data acquisition applications are usually controlled by software programs developed using various general purpose programming languages such as Assembly, BASIC, C, C++, C#, Fortran, Java, LabVIEW, etc. Stand-alone data acquisition systems are often called data loggers.

There are also open-source software packages providing all the necessary tools to acquire data from different, typically specific, hardware equipment. These tools come from the scientific community where complex experiment requires fast, flexible and adaptable software. Those packages are usually custom fit but more general DAQ packages like the Maximum Integrated Data Acquisition System can be easily tailored and is used in several physics experiment

DAS IN IOT

1.The Data acquisition system is simply defined as the system that either monitors or controls the parameters in the outside world.

2.In today's time each and every field cannot be thought of withoutconsidering these data acquisition systems. For medical instruments, industrial equipm ent, appliances for home etc. has a necessary need for these systems.

3.Hence the need of such data collection and processing the collected data for further use led to the development of Internet of Things(IOT) based Data Acquisition system using Raspberry Pi.

4.The system will be designed to collect (acquire) data from the skin response and temperature of human body.

5. The basic need was to develop our system was portability of the system, no range limitation and also more than one connectivity options for transfer of data.

6.The system will also be capable of displaying the acquired data over HDMI display. Further, it is supposed to be designed in such a way that it transfers these data to various ranges.

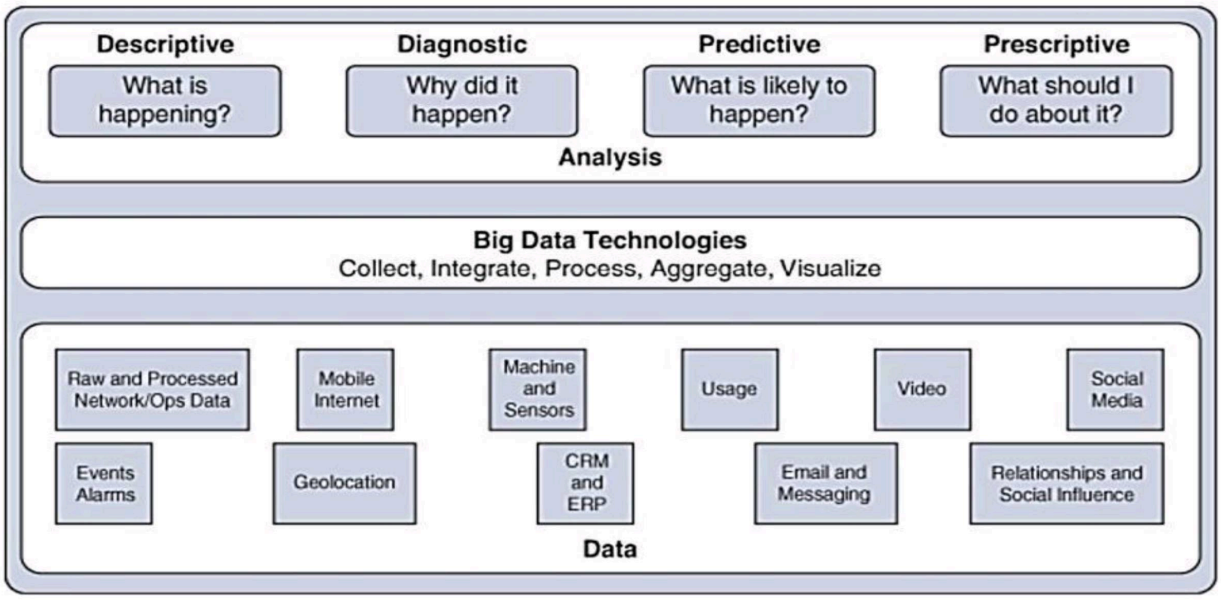
7.Thus it will be useful for data acquisition and transfer from short distances to longer. It could be regarded as a general purpose system with multiple transfer options. The system will be designed to be equipped with all possible features which enhance the working capability of the system.

8.The system will be consisted of Raspberry Pi which is one of the advanced processor and has advantages like low price and small size. Our system will consist of an Analog to Digital conversion module which will be used to convert the analog output from the sensors to digital values for further processing.

9.The Raspberry Pi also consists of SD card embedded in the same board, which will be used for storing the sensors data. The system will be using a number of protocols for achieving the concept of Internet of Things for our application.

10 The system will be able to communicate with all possible wired or wireless ways thus have multiple way of communication. Wi-Fi will be used for medium range while GSM and Ethernet will be used long range transmission. Apart from using these, a Linux based application will be provided to run on HDMI display. Support to the devices will be provided with an Android application.

UNSTRUCTURED DATA STORAGE



Descriptive: Descriptive data analysis tells you what is happening, either now or in the past. For example, a thermometer in a truck engine reports temperature values every second. From a descriptive analysis perspective, you can pull this data at any moment to gain insight into the current operating condition of the truck engine. If the temperature value is too high, then there may be a cooling problem or the engine may be experiencing too much load.

Diagnostic: When you are interested in the "why," diagnostic data analysis can provide the answer. Continuing with the example of the temperature sensor in the truck engine, you might wonder why the truck

engine failed. Diagnostic analysis might show that the temperature of the engine was too high, and the engine overheated. Applying diagnostic analysis across the data generated by a wide range of smart objects can provide a clear picture of why a problem or an event occurred.

Predictive: Predictive analysis aims to foretell problems or issues before they occur. For example, with historical values of temperatures for the truck engine, predictive analysis could provide an estimate on the remaining life of certain components in the engine. These components could then be proactively replaced before failure occurs. Or perhaps if temperature values of the truck engine start to rise slowly over time, this could indicate the need for an oil change or some other sort of engine cooling maintenance.

Prescriptive: Prescriptive analysis goes a step beyond predictive and recommends solutions for upcoming problems. A prescriptive analysis of the temperature data from a truck engine might calculate various alternatives to cost-effectively maintain our truck. These calculations could range from the cost necessary for more frequent oil changes and cooling maintenance to installing new cooling equipment on the engine or upgrading to a lease on a model with a more powerful engine. Prescriptive analysis looks at a variety of factors and makes the appropriate recommendation.

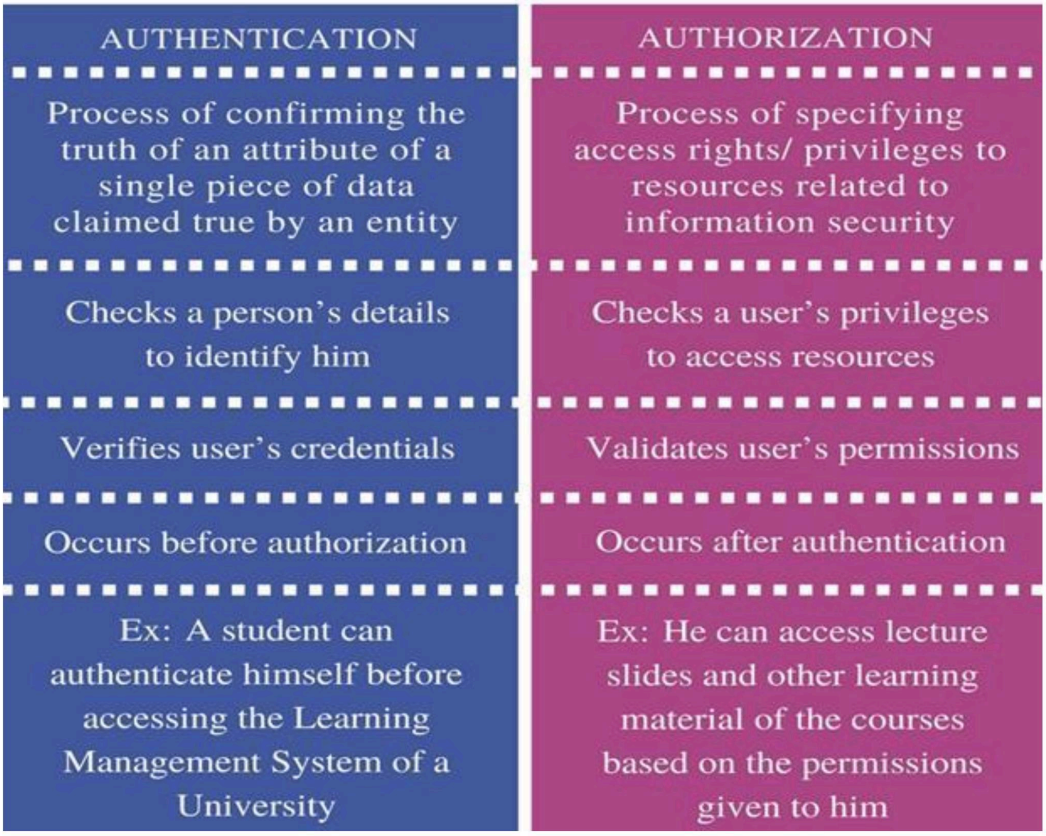
AUTHENTICATION IN IOT

Authentication and Authorization area unit utilized in respect of knowledge security that permits the safety on an automatic data system. Each area unit terribly crucial topics usually related to the online as key items of its service infrastructure. However, each the terms area unit terribly completely different with altogether different ideas. whereas it's true that they're usually employed in an equivalent context with an equivalent tool, they're utterly distinct from one another.

In authentication process, the identity of users are checked for providing the access to the system. While in authorization process, person's or user's authorities are checked for accessing the resources. Authentication is done before the authorization process, whereas authorization process is done after the authentication process.

• *Authentication* is knowing the identity of the user. For example, Alice logs in with her username and password, and the server uses the password to authenticate Alice.

• *Authorization* is deciding whether a user is allowed to perform an action. For example, Alice has permission to get a resource but not create a resource.



• Authentication in Web API

• The Web API Service assumes that the authentication process should happen in the host Server and we generally host the Web API Service at IIS. The IIS Server uses the **HTTP modules** for checking the authentication of a user. You can configure your project to use any of the built-in authentication modules which are available in IIS or ASP.NET, or you can also create your own HTTP module to perform custom authentication.

• When the host (IIS Server) authenticates the user, it generally creates a principal object (i.e. **IPrincipal** object) under which the code is going to run. So, once the Principal object (**IPrincipal** object) is created, then the host (i.e. IIS Server) attaches that **principal object** to the **current thread** by setting **Thread.CurrentPrincipal**.

• If you are confused at the moment about how the Principal object is created and how the principal object is attached to the current thread, then don't worry we will discuss all these things in greater detail in our upcoming articles. In this article, I am just going to give you an overview of how authentication and authorization happen in Web API services.

Authorization in Web API

• The Authorization Process is going to happen before executing the Controller Action Method which provides you the flexibility to decide whether you want to grant access to that resource or not.

• We can implement this in ASP.NET Web API by using the Authorization filters which will be executed before the controller action method executed. So, if the request is not authorized for that

specific resource, then the filter returns an error response to the client without executing the controller action method

CHAPTER 4
SMART TECHNOLOGY
UNDERSTANDING NETWORK CONNECTIONS IN IOT

IoT connectivity is a term defining **connection** between all the points in the **IoT** ecosystem, such as sensors, gateways, routers, applications, platforms and other systems. It usually refers to different types of **network** solutions based on their power consumption, range and bandwidth consumption.

The success or failure of Internet of Things (IoT) projects is dependent on many things, but without a reliable connection between devices, sensors and your IoT platform, your project won't even get off the ground. However, connectivity is not a matter of simply choosing a preferred wireless technology. It's equally important to understand the requirements of your application—and then choose the network technology that's the best fit.

1. Cellular

Cellular networks use the same mobile networks as smartphones to allow IoT devices to communicate. Because these networks were originally designed for power-hungry devices like smartphones, they weren't always considered the best fit for IoT devices. Eventually, the cellular industry developed new technologies that were more appropriate for IoT use cases. Today, this type of wireless network is very popular, and is considered a reliable and secure method of IoT connectivity. Cell service is available in most locations in the U.S., and this type of network covers a very large area. However, cell connectivity often isn't available in the places that most need monitoring sensors—for example, inside utility closets, elevator shafts, basements, etc. (Another IoT wireless technology class, LPWAN, might be a better fit for these locations.) And even though cellular connectivity is now less expensive and more power efficient than traditional telecom standards, cellular-connected IoT devices still require a great deal more power and energy than some other types of wireless networks.

2. Local and Personal Area Networks (LAN/PAN)

Networks that cover fairly short distances are called personal area networks (PAN) and local area networks (LAN). PAN and LAN networks are considered to be fairly cost-effective, but the transfer of data can sometimes be unreliable.

Wireless personal and local area network technologies that are commonly incorporated into IoT connectivity solutions are **WiFi** and **Bluetooth.** WiFi can be used for applications that run in a local environment, or in a distributed setting if there are multiple access points integrated into a larger network. One downside to WiFi is that it works only if the signal is strong and you're close to the access point. Also, WiFi is generally more power-hungry than people think, but it is possible to operate it in a way that's a little more power-efficient (for example, your device only connects periodically to send data, then goes back to sleep).

3. Low Power Wide Area Networks (LPWAN)

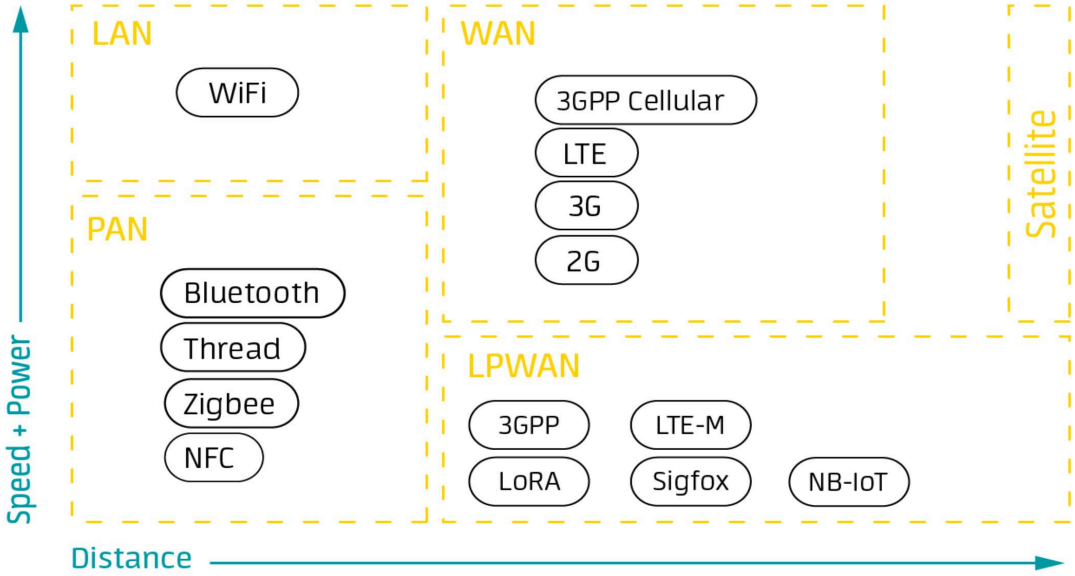
IoT devices that run on LPWANs send small packets of information infrequently and over long distances. This type of wireless network was developed in response to the early challenges of cellular connectivity. Proponents of LPWAN position it as longer-range than WiFi and Bluetooth, but using less power than cellular. Sigfox built the first LPWAN network in France and is considered the driving force behind its growth (despite the fact that Sigfox never took off in the U.S.).

A well-known and commonly used IoT network protocol in this category is **LoRaWAN** (long range wireless area network), which runs on the LoRa (long range) communication network. Advantages of LoRaWAN for IoT devices are its low power requirement (for long battery life) and relatively low-cost chipsets. Plus, under the right conditions, a single base station or gateway running on a long-range network is capable of providing service to a very large area—a few kilometers in dense urban areas and up to 15–30 kilometers in rural areas.

4. Mesh Networks

Mesh networks are best described by their connectivity configuration—how the components communicate with each other. In mesh networks, all the sensor nodes cooperate to distribute data amongst each other to reach the gateway. (A star topology, in contrast, is where all sensor nodes communicate to a central hub.)

Zigbee is one example of an IoT wireless network technology. Mesh networks are very short range and may require extra sensors throughout a building or the use of repeaters to get the coverage your application needs. Also, the nature of the way these networks communicate can result in high power consumption, especially if you need instant messaging, such as for a smart lighting application. (IoT applications that require only occasional information updates use less power.) However, mesh networks are also fairly robust, able to find the fastest and most reliable paths to send data, and easy to install, making them a popular choice for in-building use.

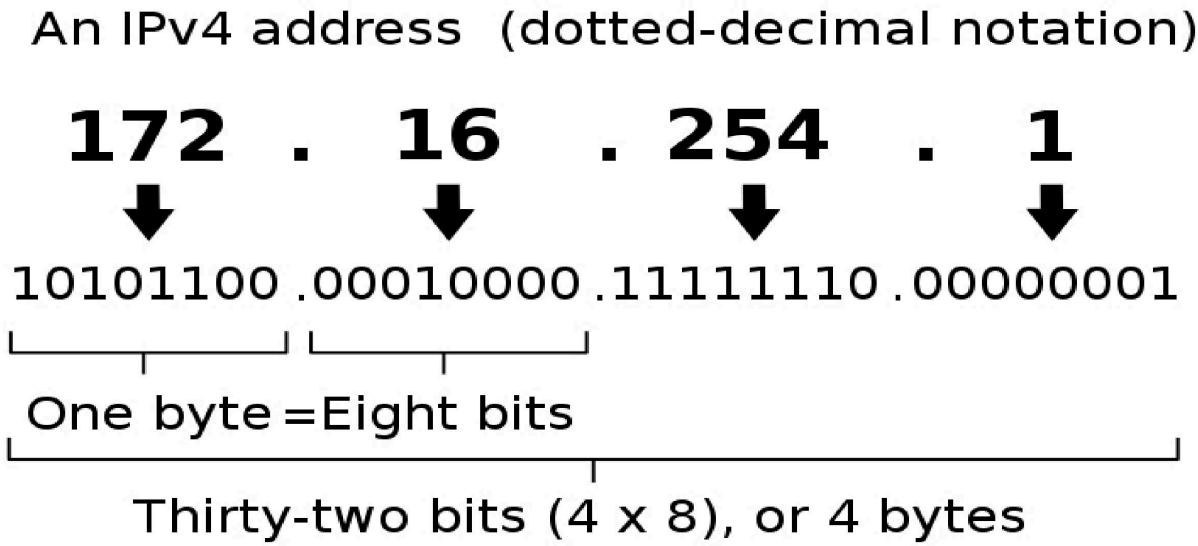


LAN VS MAN VS WAN

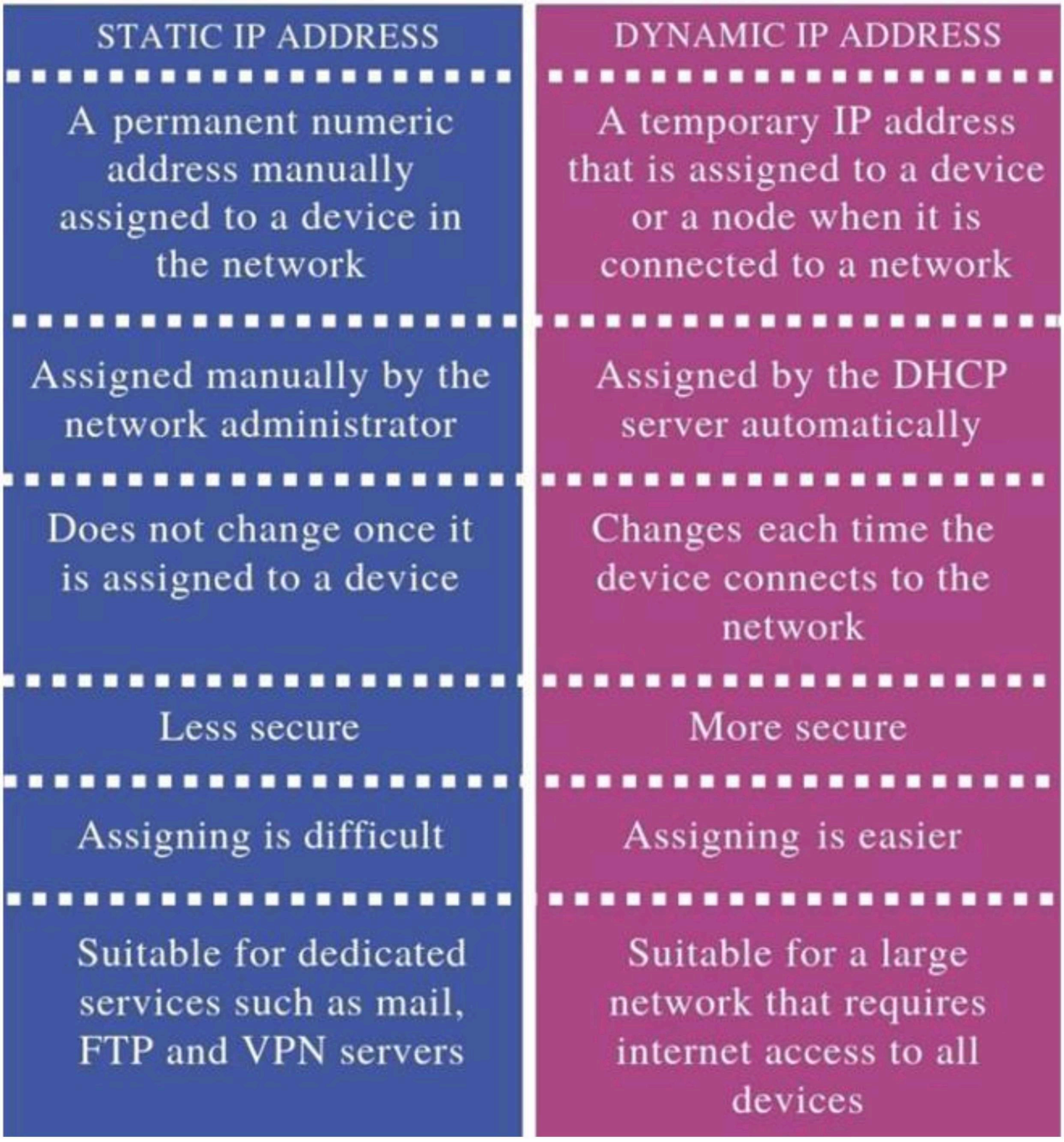
LAN	MAN	WAN
A computer network that interconnects computers within a limited area such as residence, office building, school or a laboratory	A computer network that interconnects user with computer resources in a geographical area larger than LAN but smaller than WAN	A computer network that extends over a large geographical area
Stands for Local Area Network	Stands for Metropolitan Area Network	Stands for Wide Area Network
Covers an area within 1km to 10km	Covers an area within 100km	Covers a large area that goes beyond 100km
Easier to design and maintain	Difficult and complicated to design and maintain	Difficult and complicated to design and maintain
High data transferring speed	Moderate data transferring speed	Low data transferring speed
Has a limited number of users - less congestion	Moderate congestion	Low congestion
A network in a home, school or an office	A network in a city or a small town	A network network covering a state or a country

STATIC IP ADDRESS

Static IP address is a fixed IP address assigned to the device manually. It was the original method to assign IP addresses to the devices in the network. Here, for each computer, the network administrator has to open the network configuration page and manually type the IP address. Additionally, the administrator should also include details such as subnet mask and default gateway. Moreover, this process had to be followed on all the devices in the network. Thus, it is difficult to assign static IP addresses when there are a large number of devices in the network.



However, the static IP address is not changed until it is manually changed by the network administrator or the Internet Service Provider (ISP). Also, this address does not change each time when the user connects to the network. In other words, the device always uses the same IP address to connect to the internet. Therefore, the static IP addressing provides multiple advantages. It gives minimum downtime while providing remote access. Also, the user can access the device from any location.



A dynamic IP address is an address obtained from a Dynamic Host Configuration Protocol (DHCP) server. It assigns a device with dynamic IP address, subnet mask, default gateway, and a DNS server. In a Microsoft computer, selecting the option "obtain an IP address automatically" in the network property window will set the device to obtain an IP address dynamically.

The dynamic IP address changes frequently. Each time the device connects to the network, the dynamic IP address changes. When the device tries to connect to the

network, the DHCP server provides a dynamic address. When the user types a URL on the web browser, the DNS server maps the domain name to the IP address. Overall, Dynamic IP addressing is automatic and it makes managing a network easier.

Types of IP addresses

There are different categories of IP addresses, and within each category, different types.

Consumer IP addresses

Every individual or business with an internet service plan will have two types of IP addresses: their private IP addresses and their public IP address. The terms public and private relate to the network location — that is, a private IP address is used inside a network, while a public one is used outside a network.

Private IP addresses

Every device that connects to your internet network has a private IP address. This includes computers, smartphones, and tablets but also any Bluetooth-enabled devices like speakers, printers, or smart TVs. With the growing internet of things, the number of private IP addresses you have at home is probably growing. Your router needs a way to identify these items separately, and many items need a way to recognize each other. Therefore, your router generates private IP addresses that are unique identifiers for each device that differentiate them on the network.

Public IP addresses

A public IP address is the primary address associated with your whole network. While each connected device has its own IP address, they are also included within the main IP address for your network. As described above, your public IP address

is provided to your router by your ISP. Typically, ISPs have a large pool of IP addresses that they distribute to their customers. Your public IP address is the address that all the devices outside your internet network will use to recognize your network.

Public IP addresses

Public IP addresses come in two forms – dynamic and static.

Dynamic IP addresses

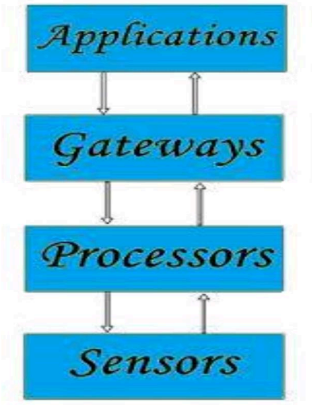
Dynamic IP addresses change automatically and regularly. ISPs buy a large pool of IP addresses and assign them automatically to their customers. Periodically, they re-assign them and put the older IP addresses back into the pool to be used for other customers. The rationale for this approach is to generate cost savings for the ISP. Automating the regular movement of IP addresses means they don’t have to carry out specific actions to re-establish a customer's IP address if they move home, for example. There are security benefits, too, because a changing IP address makes it harder for criminals to hack into your network interface.

Static IP addresses

In contrast to dynamic IP addresses, static addresses remain consistent. Once the network assigns an IP address, it remains the same. Most individuals and businesses do not need a static IP address, but for businesses that plan to host their own server, it is crucial to have one. This is because a static IP address ensures that websites and email addresses tied to it will have a consistent IP address — vital if you want other devices to be able to find them consistently on the web.

BUILDING BLOCKS OF IOT

Building Blocks Of IoT Four things form basic building blocks of the IoT system –sensors, processors, gateways, applications. Each of these nodes has to have its own characteristics in order to form an useful IoT system.



Sensors: These form the front end of the IoT devices. These are the so-called "Things" of the• system. Their main purpose is to collect data from its surroundings (sensors) or give out data to its surrounding (actuators). These have to be uniquely identifiable devices with a unique IP address so that they• can be easily identifiable over a large network. These have to be active in nature which means that they should be able to collect• real-time data. These can either work on their own (autonomous in nature) or can be made to work by the user depending on their needs (user-controlled). Examples of sensors are gas sensor, water quality sensor, moisture sensor, etc.• Processors: Processors are the brain of the IoT system. Their main function is to process the data• captured by the sensors and process them so as to extract the valuable data from the enormous amount of raw data collected. In a word, we can say that it gives intelligence to the data.

Processors mostly work on real-time basis and can be easily controlled by• applications. These are also responsible for securing the data – that is performing encryption and decryption of data. Embedded hardware devices, microcontroller, etc are the ones that process the data• because they have processors attached to it. Gateways:

Gateways are responsible for routing the processed data and send it to proper• locations for its (data) proper utilization. In other words, we can say that gateway helps in to and fro communication of the• data. It provides network connectivity to the data. Network connectivity is essential for any IoT

system to communicate. LAN, WAN, PAN, etc are examples of network gateways.• Applications: Applications form another end of an IoT system.

Applications are essential for proper utilization of all the data collected. These cloud-based applications which are responsible for rendering the effective• meaning to the data collected. Applications are controlled by users and are a delivery point of particular services. Examples of applications are home automation apps, security systems, industrial• control hub, et

UNDERSTANDING IP ADDRESS IN IOT

An IP address is a unique address that identifies a device on the internet or a local network. IP stands for "Internet Protocol," which is the set of rules governing the format of data sent via the internet or local network.

In essence, IP addresses are the identifier that allows information to be sent between devices on a network: they contain location information and make devices accessible for communication. The internet needs a way to differentiate between different computers, routers, and websites. IP addresses provide a way of doing so and form an essential part of how the internet works.

How do IP addresses work

If you want to understand why a particular device is not connecting in the way you would expect or you want to troubleshoot why your network may not be working, it helps understand how IP addresses work.

Internet Protocol works the same way as any other language, by communicating using set guidelines to pass information. All devices find, send, and exchange information with other connected devices using this protocol. By speaking the same language, any computer in any location can talk to one another.

The use of IP addresses typically happens behind the scenes. The process works like this:

Your device indirectly connects to the internet by connecting at first to a network connected to the internet, which then grants your device access to the internet.

When you are ~~at~~ home, that network will probably be your Internet Service Provider (ISP). At work, it will be your company network.

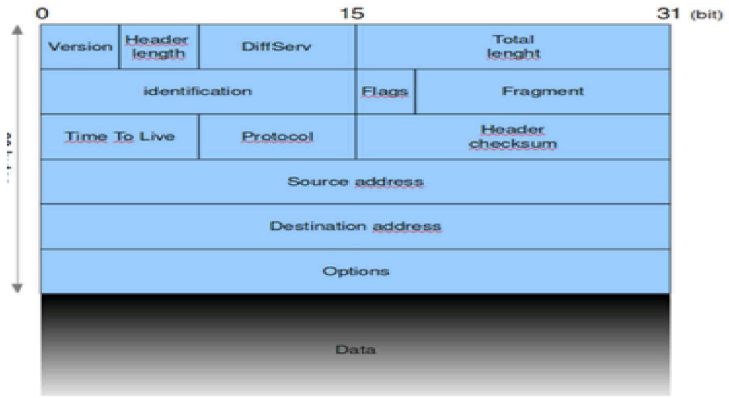
Your IP address is assigned to your device by your ISP.

Your internet activity goes through the ISP, and they route it back to you, using your IP address. Since they are giving you access to the internet, it is their role to assign an IP address to your device.

However, your IP address can change. For example, turning your modem or router on or off can change it. Or you can contact your ISP, and they can change it for you.

When you are out and about – for example, traveling – and you take your device with you, your home IP address does not come with you. This is because you will be using another network (Wi-Fi at a hotel, airport, or coffee shop, etc.) to access the internet and will be using a different (and temporary) IP address, assigned to you by the ISP of the hotel, airport or coffee shop.

IP4



Version – It is a 4-bit field that describes the IP type that is being used.

Header Length – It is a 4-bit field that gives the length of the IPv4 header in 32-bit words.

DiffServ – It is an 8-bit field that represents precedence, delay, throughput, reliability etc. Moreover, It is the Type of Service (ToS) field.

Total Length – It is a 16-bit field that describes the whole length of the packet.

Identification – It is a 16-bit field. When a particular packet belongs to a sequence of packets, all of them gets the same identification number. This helps to recognize them at the receiving end.

Flag – It is a 3-bit field that explains the fragmentation options.

Fragment – It indicates the fragment to which the packet belongs.

Time To Live (TTL) – It is an 8-bit field that indicates the time in seconds or number of router hops the packet can have before discarding.

Protocol – It is an 8-bit field that describes the protocol of receiving the data payload.

Header Checksum – It helps to verify the validity of the header.

Source IP address – It is a 32-bit address that describes the address of the device that sends the packet.

Destination IP address – It is a 32-bit address that describes the address of thereceiving end.

Options – It is used for tasks such as testing, security, etc.

Data – It represents the real data that should be transmitted.

A packet in a network that uses IPv6 creates an IPv6 header. It is as follows.

Version – It is a 4-bit field that describes the IP type that is being used.

Traffic class – It is an 8-bit field that describes the packet's class or priority. Moreover, it is similar to the IPv4 ToS field.

Flow label – It is 20-bit long. Moreover, it indicates the position of the packet in a set of packets and helps to prioritize the packets, especially when transmitting voice.

Payload length – It is 16-bit long and displays the length of IPv6 payload with the extension headers and upper layer protocol data.

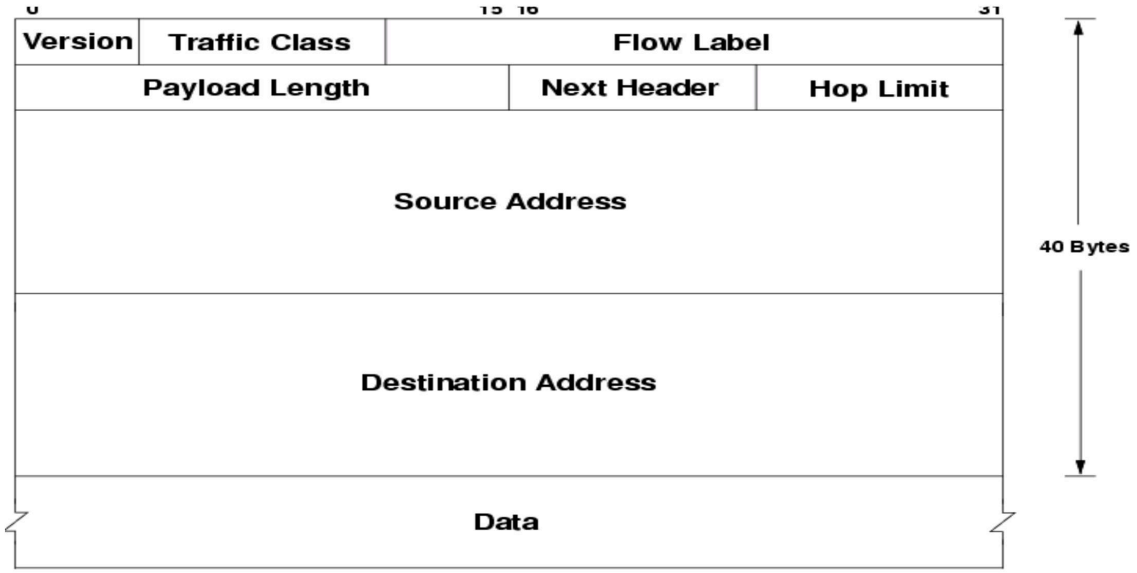
Next Header – It is an 8-bit field that shows the type of the first extension or the protocol in the upper layer.

Hop Limit – It is 8-bit long. It indicates the maximum number of routers the packet is capable of passing. This is similar to TTL field in IPv4 header.

Source address – It is 128 bits long. It is the address of the device that sends the packet.

Destination address – It is also 128 bits long. It is the address of the device that receives the packet.

Data – It represents the real data that should be transmitted.



CHAPTER 5

SMART TV

SMART TV AND ITS USE:

A **smart TV**, also known as a **connected TV** (**CTV**), is a traditional television set with integrated Internet and interactive Web 2.0 features, which allows users to stream music and videos, browse the internet, and view photos. Smart TVs are a technological convergence of computers, televisions, and digital media players. Besides the traditional functions of television sets provided through traditional broadcasting media, these devices can provide access to over-the-top media services such as streaming television and internet radio, along with home networking access Smart TV should not be confused with Internet TV, IPTV, or streaming television. *Internet TV* refers to receiving television content over the Internet instead of traditional systems such as terrestrial, cable, and satellite, regardless of how the Internet is delivered. IPTV is one of the Internet television technology standards for use by television broadcasters. *Streaming television* is a term used for programs created by a wide variety of companies and individuals for broadcast on Internet TV.

The concept of the smart TV isn't particularly new. Smart TVs have been around since 2007 or so, under many different labels, including "connected" TV, "hybrid" TV, "IPTV," and "Internet" TV. (One could even argue that the concept has actu- ally been around since 1995's WebTV box, which served as an Internet client for traditional TVs.)

What's Inside a Smart TV?

At its most basic, a smart TV is a television set that can connect to and interact with the Internet. In practical terms, that means the television must include the following:

Wi-Fi radio or Ethernet connection, for connecting to your home network.

Central processing unit (CPU), the computer brain that manages all the device's operations and commands.

Operating system (OS) that serves as the interface between the CPU and software-based applications.

Graphical user interface (GUI) for displaying menus and other options.

Software-based apps that enable connection to various web-based ser- vices. For example, a smart TV might have built-in apps for Netflix, Hulu, and Pandora. Most smart TVs come with several apps pre- installed; some smart TVs enable additional apps to be installed after purchase.

Some smart TVs also include apps and associated technologies that enable the device to play back media stored on your home network. In some cases, this capability is built into the OS, as with the Apple TV; in other cases, this capability is enabled by DNLA or UPnP compatibility .

Some smart TVs include a built-in camera and microphone, like the one shown in Figure 3.1, for connecting with video-sharing and chat services, such as Skype. Some more advanced smart TVs use the built-in camera/microphone to navigate the onscreen menus, via a series of hand gestures or voice commands.

Naturally, a smart television set (not a set-top box) will also include a traditional television tuner for viewing broadcast, cable, or satellite programming. You typi- cally switch from the normal viewing screen to a GUI menu for the web-based services and apps.

All smart TVs are controlled by some sort of remote control. Some remotes are basic affairs, with just enough buttons to navigate the onscreen menus. Others include keyboards (useful for typing in search terms), trackpads, even game controllers. Most smart TVs can be controlled by universal remotes, such as those in the Logitech Harmony line. Some smart TVs can be controlled by smartphone or tablet apps.

Remember, too, that a smart TV doesn't have to be a literal TV. A smart TV device, like the aforementioned Roku box, contains the same circuitry and apps as a literal smart TV, but without the TV part. Instead, the set-top box connects to a regular TV (typically via high-definition multimedia interface [HDMI]), enabling the TV to display media played on the external device.

What You Need to Use a Smart TV

Right out of the box, a smart TV has little or no functionality. To utilize all the fea- tures of a smart TV, you need to provide the following:

An Internet connection.

A home network that interfaces with your Internet connection. This can be a wireless (Wi-Fi) or wired (Ethernet) network.

Electricity. Duh.

If you have a smart TV set-top box, you'll also need an HDMI cable to connect the device to your traditional television set.

What a Smart TV Does

So a smart TV is a TV or set-top box that integrates Internet capabilities. What exactly does that mean?

Most smart TVs can perform the following functions:

Connect to the Internet via a local network. That means connecting to your home network and sharing your Internet connection. Most smart TVs connect via Wi-Fi, although some can connect via Ethernet.

Play video content from web-based streaming video services, such as Netflix, Hulu Plus, and Amazon Instant Video.

Play music from web-based streaming audio services, such as Pandora and Spotify.

Play digital media stored on other devices connected to your home network.

Access selected websites and web-based services, such as Facebook, Twitter, and AccuWeather. Some smart TVs offer full-fledged web browsers, although it's more common to find discrete apps for specific sites and services.

Smart TV Operating Systems

All smart TVs and smart TV devices are like mini computers, in that they include a built-in OS and the appropriate software or middleware to run the necessary apps. Now, these devices don't run a full-blown consumer OS, such as Windows, but rather smaller, more stripped down OS's developed specifically for these purposes.

There are a number of smart TV OS's in use today, many proprietary to a specific company or device. These include the following:

Android TV, used in the Google Chromecast and selected Sony smart TVs
Fire OS, used in Amazon's streaming devices
Firefox OS, used on Panasonic devices
iOS, Apple's mobile OS used in the Apple TV box (and iPhones and iPads, of course)
Roku OS, used by Roku
Tizen, a Linux-based OS used by Samsung
webOS, a Linux derivative used by LG

This proliferation of OS's means that no two brands of smart TVs look or work exactly alike. While all these OS's do pretty much the same thing, they do it all differently; every company puts its own spin on onscreen menus, navigation, and operation. For this reason, you want to spend some time with a given interface when you're shopping for a smart TV or device.

IntegratingSmartTVsintotheInternetofThings

Okay, so it's pretty obvious that the current generation of smart TVs has very little to connect it to the Internet of Things. Just because a TV or set-top device lets you watch both broadcast and Internet-based programming doesn't make it hyper- intelligent or even moderately clever. It just adds more types of programming to what is still more or less a non-participatory device. A TV that can play old episodes of *Doctor Who* on Netflix is still just a TV.

For a smart TV to become truly smart, it needs to do more. Not surprisingly, there are people working on this.

The first thing smart TV manufacturers are likely to do is make it easier to control the smart TVs themselves. Let's face it, picking through the choices on Hulu or searching for your favorite movie on Netflix isn't easily accomplished with a traditional four-arrow remote control. Some manufacturers have experimented with including a full-fledged keyboard in a handheld remote, but that's a little too cum- be some. A better solution might be a touch screen tablet-like controller, a remote app on a Smartphone or iPad, or even Siri-like voice control. Samsung, if you recall, uses its built-in camera to enable rudimentary gesture commands, which is another way to go. Whatever the approach, the smart TV companies need to make it easier to find all the various programming they enable

future generations of smart TVs are likely to get smarter about what you like to watch. These new smart TVs will collect data about what you watch and when (and, if you have multiple viewers in the same household, which you probably do, what each viewer likes to watch) and make assumptions about your future viewing habits. Even better, your smart TV might connect to your Face book or Twitter account to discover what shows your friends are watching.

All this data will be assembled and collated, and your smart TV will start mak- ing recommendations for future viewing. The set might even go the next step and create a new "just for you" screen with one-click access to the recommended programming, or just set the onboard DVR to record these programs for your viewing convenience. With your smart TV making smart choices about what you want to watch, you'll no longer have to deal with the increasingly Byzantine program guide. You won't have to think about what you want to watch at all; your smart TV will do your thinking for you.

Future smart TVs may also use their Internet connectivity to overlay related infor- mation on the main viewing screen. If you're watching a sporting event, for exam- ple, you may see team or player stats superimposed on the screen, or displayed in a side window. If you're viewing a classic movie, you might see bios of the director and stars, with links to other similar movies you might like.

In addition, expect smart TVs to include more interactive chat capabilities. When you're watching a movie or show, you'll be able to tweet or post to Facebook about what you're watching, and participate in group chats about the show. These might be video chats, conducted in a pop-up window and enabled by your set's built-in camera.

Future iterations of smart TV will turn the TV set into a hub for a variety of household activities. For example, you might feed video from your home's security cameras to your smart TV,

CHAPTER 6
CASE STUDY OF IOT DEVICES

Definition: "A case study is a research strategy and an empirical inquiry that investigates a phenomenon within its real-life context. Case studies are based on an in-depth investigation of a single individual, group or event to explore the causes of underlying principles". One of the most promising IoT use cases is creating smarter, more efficient cities. Public energy grids can be optimized to balance workloads, predict energy surges, and distribute energy more equitably to customers. Traffic lights could be synced using IoT to adapt to traffic conditions in real-time.

IoT is the next step in the evolution of the internet and is being used in about everything you can think of.

IoT Applications: Smart Home, Smart Buildings and Infrastructure IoT home automation is the ability to control domestic appliances by electronically controlled, internet-connected systems. It may include setting complex heating and lighting systems in advance and setting alarms and home security controls, all connected by a central hub and remote-controlled by a mobile app.

Applications of home automation

Rebuilding consumer expectations, home automation has been projected to target wide array applications for the new digital consumer. Some of the areas where consumers can expect to see home automation led IoT-enabled connectivity are:

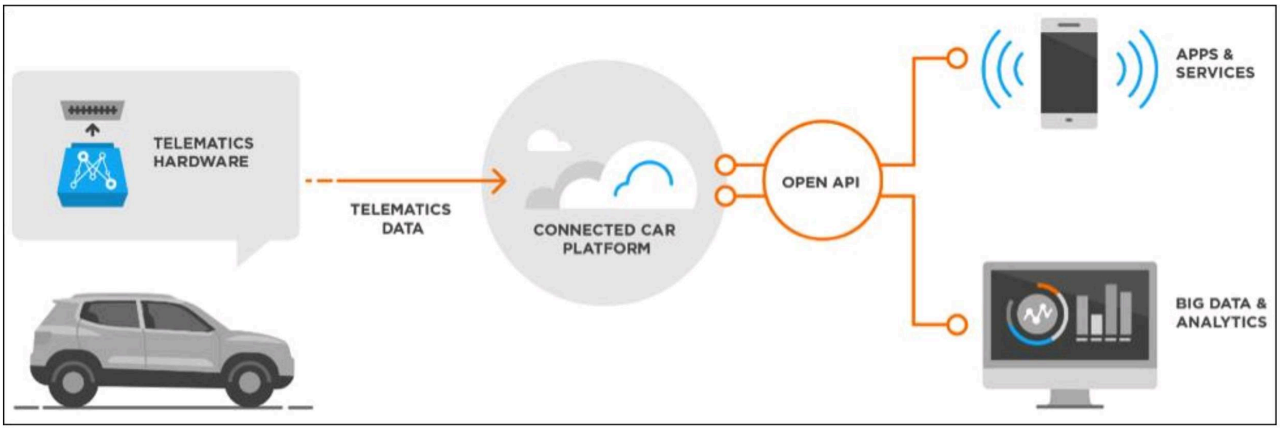
Lighting control
HVAC
Lawn/Gardening management
Smart Home Appliances
Improved Home safety and security
Home air quality and water quality monitoring
Natural Language-based voice assistants
Better Infotainment delivery
AI-driven digital experiences
Smart Switches
Smart Locks
Smart Energy Meters

Major IoT platform as a service provider:
AWS IoT
Azure IoT
Thingworx
Ubidots

Thingspeak
Carriots
Konekt
TempoIQ
Xively
IBM Bluemix

Applications of iot in smart car

Imagine a world where your car not only drives itself, but also says intelligent things. This would look like an impossibility about five years ago, but today the IoT is already breaking fresh ground for tech companies and car manufacturers, enabling them to realize their idea of a 'connected car.'



Behavioral Data — Mojio's telematics device gathers information about speed, steering, and braking inputs to determine driver's fatigue level and issue alerts. Long-term driving behavior data can also be used to help the user adopt a more fuel efficient driving style and calculate risk by insurance companies.

Diagnostic Data — With the ability to access vehicle's data remotely, car manufacturers can assess the health of a vehicle and combine this capability with in-car voice communication to notify customers when service is required.

Contextual Data — Led by Google and Amazon, contextual targeting of advertisements based on the search data of an individual has become a usual practice in the digital world. Mojio is using the same principle to offer more personalized advice to car drivers. It enriches the behavioral and contextual data of a customer with geolocation data, posted speed limits, and updated traffic flow conditions to provide valuable recommendations to the driver.

IoT in Healthcare

Contact tracing
Pathogen detection
Thermal detection (elevated temperature)
No-touch sanitation dispensers
Automated hand hygiene
Hygiene monitoring
Workspace and floor sanitation
Air quality sensors

Biometrics scanners
Vital signs monitoring
Remote patient communications
Instrument sterilization
Medication dispensing

SMART CITIES

Road traffic. Smart cities ensure that their citizens get from point A to point B as safely and efficiently as possible. ...
Smart parking. ...
Public transport. ...
Utilities. ...
Street lighting. ...
Waste management. ...
Environment. ...
Public safety.
