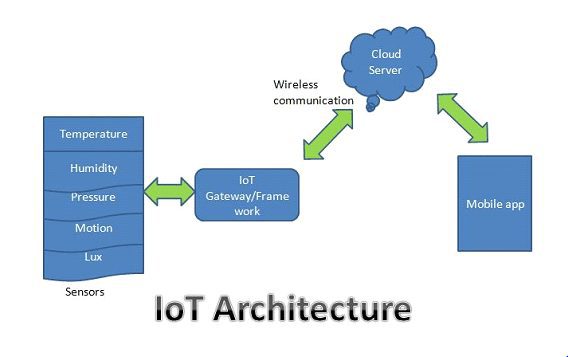
**INTERNET OF THINGS**

DEFINITION

This is the concept of connecting any device with an on and off switch to the Internet (and/or to each other). The IoT is a giant network of connected "things" (which also includes people).  The relationship will be between people-people, people-things, and things-things.

ARCHITECTURE OF IOT



IOT ARCHITECTURE LAYERS

1. The client side (IoT Device Layer)

2. Operators on the server side (IoT Getaway Layer)

3. A pathway for connecting clients and operators (IoT Platform Layer)

4 STAGES OF IOT ARCHITECTURE

1. Sensors and actuators

Sensing is for collecting various data from related objects and sending it to a database, data warehouse or data centre. The gathered data is further analysed to perform specific actions based on required services. The sensors can be humidity sensors, temperature sensors, wearable sensing devices, mobile phones and many others.

1. Internet getaways and Data Acquisition Systems

Communication technologies connect heterogeneous objects together to offer specific services. The communication protocols available for the IoT are: Wi-Fi, Bluetooth, IEEE 802.15.4, Z-wave, LTE-Advanced, Near Field Communication, ultra-wide bandwidth, Low-Power Wide-Area Network and emerging standards.

Computation, the hardware processing units, like microcontrollers, microprocessors, system on chips (SoCs) or field programmable gate arrays, and software applications perform this task. Many hardware platforms like Arduino or Raspberry PI are developed and various software platforms are utilized. The cloud platform is a particularly important computational part of IoT, since it is very powerful in processing various data in real time and extracting all kinds of valuable information from the gathered data.

1. Data center and cloud.

A Data center can be defined as a facility which incorporates components such as servers, communication media and data storage facilities. Along with this it also contains various components which are essential to run a data center like power supply, backup systems, redundant communication connection, HVAC systems, security devices etc. It’s an on-premise hardware solution where all the resources are locally present at access which is typically run and maintained by in-house.

On the other hand, a Cloud is a virtual infrastructure that is accessed or delivered with a local network or accessing the remote location through internet. The cloud services can be accessed on-demand whenever the user requires on a pay per use basis or a dedicated resource, this model is known as Infrastructure as a Service (IaaS). Within this environment, the user can access computing resources, networking services and storage which the users can access on-demand without any requirement of physical infrastructure. It is an Off-premise form of computing which can be accessed from the internet, it’s maintenance and updates is maintained and controlled by the third-party

Elements make up the IoT are three categories :

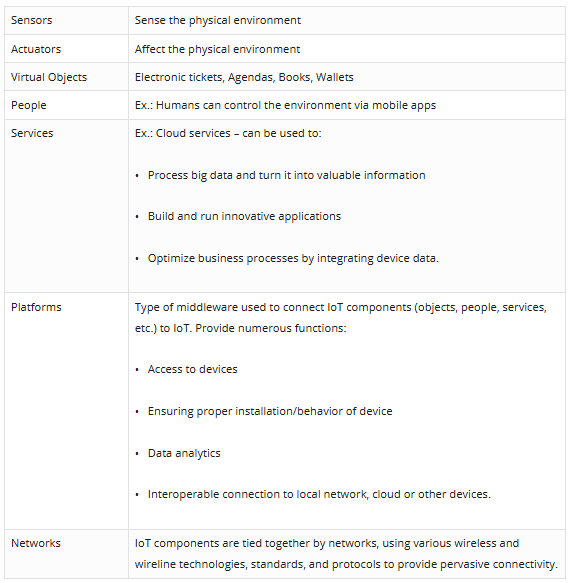
* Hardware;
* Software; and
* Cloud.

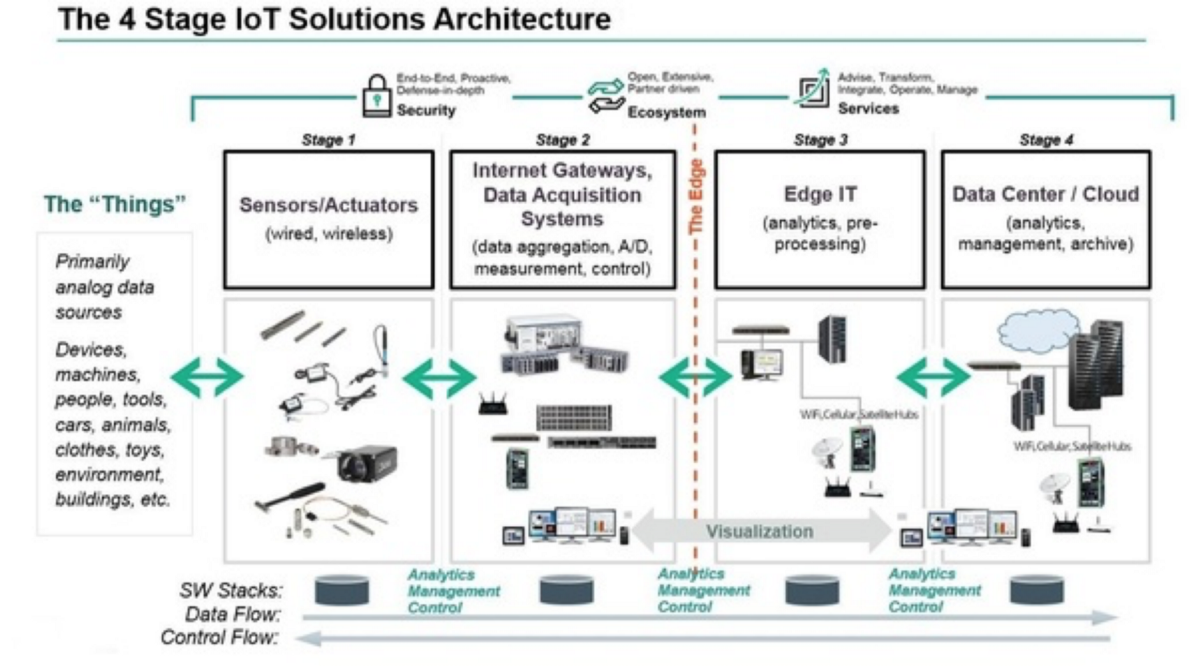
Common elements found in IoT hardware include:

* Low energy sensors;
* Communication services – gateways, modems, routers; and
* Touch screens and battery support/power.

Hardware trends include decreasing the size of devices and moving to a system that does not require a battery. Once data from sensors is collected, that information needs to be filtered and sent to the end user. That may be a consumer or a commercial or industrial user. It also may be another device in the machine-to-machine workflow. Once the information is provided, it can be presented in visualizations on mobile devices.

And of course there is cloud. Cloud solutions allow large, complex processes to be completed away from the device. Gathered data is transmitted to a cloud-based service where the information coming in from the IoT device is aggregated with other cloud based data to provide useful information for the end user. The data being consolidated can be information from other internet sources as well as from others subscribing with similar IoT devices. Most often, there will be some data processing required to provide useful information that is not necessarily obvious in the raw data.





**Stage 1. Networked things (wireless sensors and actuators)**

The outstanding feature about sensors is their ability to convert the information obtained in the outer world into data for analysis.

For actuators, the process goes even further — these devices are able to intervene the physical reality. For example, they can switch off the light and adjust the temperature in a room.

**Stage 2. Sensor data aggregation systems and analog-to-digital data conversion**

Even though this stage of IoT architecture still means working in a close proximity with sensors and actuators, Internet getaways and data acquisition systems (DAS) appear here too. Specifically, the later connect to the sensor network and aggregate output, while Internet getaways work through Wi-Fi, wired LANs and perform further processing.

The vital importance of this stage is to process the enormous amount of information collected on the previous stage and squeeze it to the optimal size for further analysis. Besides, the necessary conversion in terms of timing and structure happens here.

In short, Stage 2 makes data both digitalized and aggregated.

**Stage 3. The appearance of edge IT systems**

During this moment among the stages of IoT architecture, the prepared data is transferred to the IT world. In particular, edge IT systems perform enhanced analytics and pre-processing here. For example, it refers to machine learning and visualization technologies. At the same time, some additional processing may happen here, prior to the stage of entering the data center.

Likewise, Stage 3 is closely linked to the previous phases in the building of an architecture of IoT. Because of this, the location of edge IT systems is close to the one where sensors and actuators are situated, creating a wiring closet. At the same time, the residing in remote offices is also possible.

**Stage 4. Analysis, management, and storage of data**

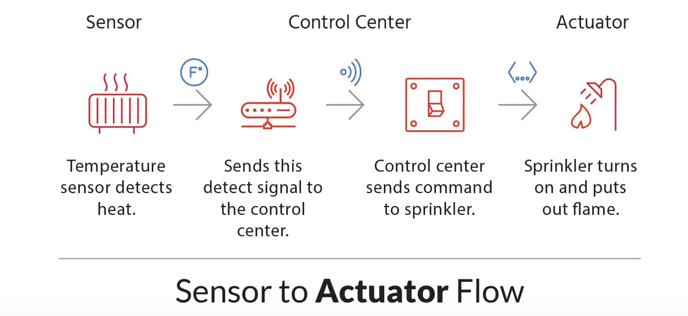
The main processes on the last stage of IoT architecture happen in data center or cloud. Precisely, it enables in-depth processing, along with a follow-up revision for feedback. Here, the skills of both IT and OT (operational technology) professionals are needed. In other words, the phase already includes the analytical skills of the highest rank, both in digital and human worlds. Therefore, the data from other sources may be included here to ensure an in-depth analysis.

After meeting all the quality standards and requirements, the information is brought back to the physical world — but in a processed and precisely analyzed appearance already.

PHYSICAL DEVICES IN IOT

Actuators and sensors are devices that enable interaction with the physical world.   
Example, Moti is an actuator. Moti creates smart motors and apps for robots. Attach the smart motor to anything, add power, and Moti gives you the ability to control the item from your desktop browser.

Actuators are devices that are used to manipulate the physical environment, such as the temperature control valves used in smart homes. Actuators take electrical input and transforms the input into tangible action. These technologies collect a high amount of data, which can be very valuable and useful to an enterprise once it has been stored, organized, and processed.



**Sensors:**

A **sensor** is a device that detects and responds to some type of input from the physical environment. The specific input could be light, heat, motion, moisture, pressure, or any one of a great number of other environmental phenomena.

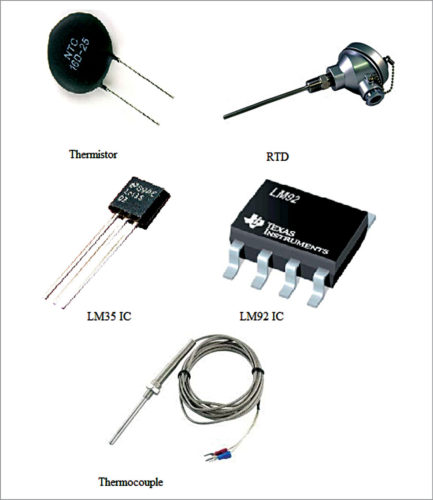
**Some common types of IoT sensors**

**Temperature sensors**

These devices measure the amount of heat energy generated from an object or surrounding area. They find application in air-conditioners, refrigerators and similar devices used for environmental control. They are also used in manufacturing processes, agriculture and health industry.

Temperature sensors can be used almost in every IoT environment, from manufacturing to agriculture. In manufacturing, sensors are used to monitor the temperature of machines. In agriculture, these can be used to monitor the temperature of soil, water and plants.

Temperature sensors include thermocouples, thermistors, resistor temperature detectors (RTDs) and integrated circuits (ICs).



Temperature sensors

[**Humidity sensors**](https://electronicsforu.com/resources/electronics-components/humidity-sensor-basic-usage-parameter)

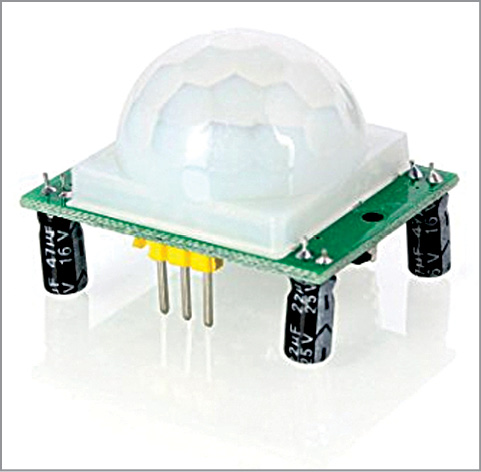
The amount of water vapour in air, or humidity, can affect human comfort as well as many manufacturing processes in industries. So monitoring humidity level is important. Most commonly used units for humidity measurement are relative humidity (RH), dew/frost point (D/F PT) and parts per million (PPM).



humidity sensor

**Motion sensors**

Motion sensors are not only used for security purposes but also in automatic door controls, automatic parking systems, automated sinks, automated toilet flushers, hand dryers, energy management systems, etc. You use these sensors in the IoT and monitor them from your smartphone or computer. HC-SR501 passive infrared (PIR) sensor is a popular motion sensor for hobby projects.



PIR motion sensor

**Gas sensors**

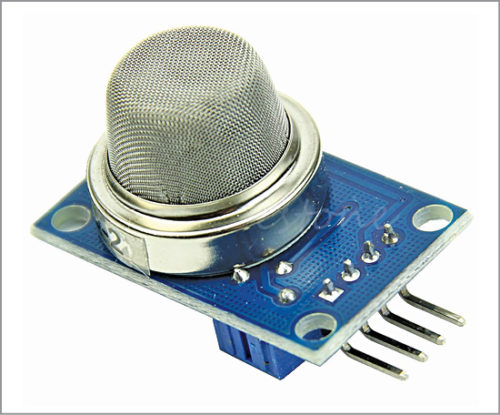
These sensors are used to detect toxic gases. The sensing technologies most commonly used are electrochemical, photo-ionisation and semiconductor. With technical advancements and new specifications, there are a multitude of gas sensors available to help extend the wired and wireless connectivity deployed in IoT applications.



Gas sensors

**Smoke sensors**

Smoke detectors have been in use in homes and industries for quite a long time. With the advent of the IoT, their application has become more convenient and user-friendly. Furthermore, adding a wireless connection to smoke detectors enables additional features that increase safety and convenience.



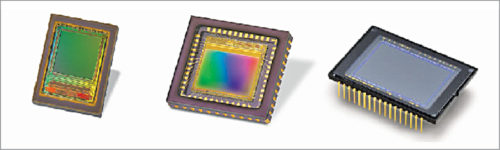
Arduino-compatible smoke sensor

[**Pressure sensors**](https://electronicsforu.com/resources/learn-electronics/pressure-sensors-market)

These sensors are used in IoT systems to monitor systems and devices that are driven by pressure signals. When the pressure range is beyond the threshold level, the device alerts the user about the problems that should be fixed. For example, BMP180 is a popular digital pressure sensor for use in mobile phones, PDAs. Pressure sensors are also used in smart vehicles and aircrafts to determine force and altitude, respectively. In vehicle, tyre pressure monitoring system (TPMS) is used to alert the driver when tyre pressure is too low and could create unsafe driving conditions.

**Image sensors**

These sensors are found in digital cameras, medical imaging systems, night-vision equipment, thermal imaging devices, radars, sonars, media house and biometric systems. In the retail industry, these sensors are used to monitor customers visiting the store through IoT network. In offices and corporate buildings, they are used to monitor employees and various activities through IoT networks.



Different types of image sensors

[**Accelerometer sensors**](https://electronicsforu.com/resources/learn-electronics/accelerometer-tilt-sensor)

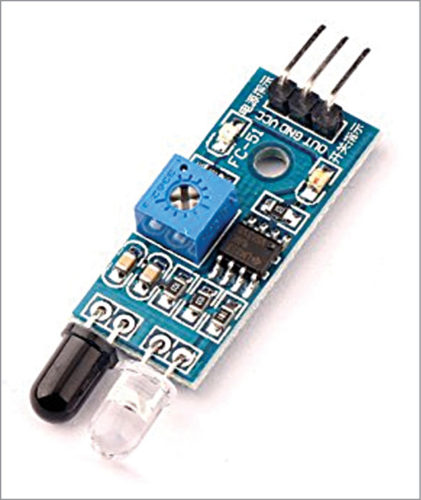
These sensors are used in smartphones, vehicles, aircrafts and other applications to detect orientation of an object, shake, tap, tilt, motion, positioning, shock or vibration. Example: piezoelectric accelerometers.



Various types of accelerometer sensors

**IR sensors**

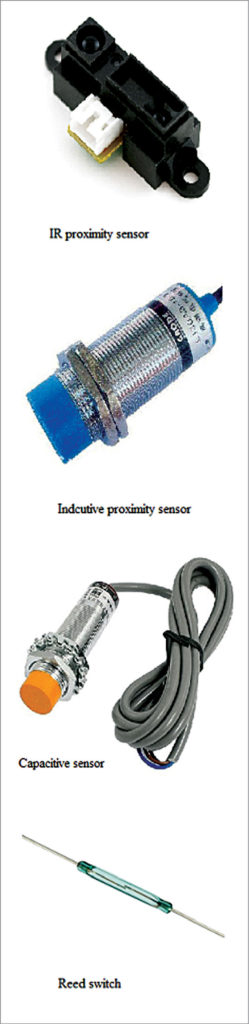
These sensors can measure the heat emitted by objects. They are used in various IoT projects including healthcare to monitor blood flow and blood pressure, smartphones to use as remote control and other functions, wearable devices to detect amount of light, thermometers to monitor temperature and blind-spot detection in vehicles.



IR sensor

**Proximity sensors**

These sensors detect the presence or absence of a nearby object without any physical contact. Different types of proximity sensors are inductive, capacitive, photoelectric, ultrasonic and magnetic. These are mostly used in object counters, process monitoring and control.



# Liquid Level Sensors

Reventec is a world-leader in the design and development of capacitive liquid level sensing technology. Our solid-state capacitive liquid level sensors are suitable for monitoring fuel level, oil level, water and coolant level, as well as other liquids.

### Heartbeat Sensor

The heartbeat sensor is based on the principle of photo phlethysmography. It measures the change in volume of blood through any organ of the body which causes a change in the light intensity through that organ (a vascular region). In case of applications where heart [pulse rate is to be monitored](https://www.edgefxkits.com/patient-health-monitoring-system-with-location-details-by-gps-over-gsm), the timing of the pulses is more important. The flow of blood volume is decided by the rate of heart pulses and since light is absorbed by blood, the signal pulses are equivalent to the heart beat pulses.

ACTUATORS

Actuators form an integral part of so many different pieces of equipment.

An **actuator** is a component of a machine that is responsible for moving and controlling a mechanism or system, for example by opening a valve. In simple terms, it is a "mover". ... When it receives a control signal, an **actuator** responds by converting the signal's energy into mechanical motion.

Examples: Security systems are a good example of this. Since linear actuators are used in many systems to extend the use and range of a camera, the Internet of Things can use it in a variety of ways like having the sensors of the system tripped can mean that the Internet of Things is triggered to send information from the security system to a mobile device of some kind;to warn the appropriate people about the possible security breach. The Internet of Things (having detected a possible problem), can take control of the actuators and turn the camera to where it needs to go, to see what is happening more clearly.

Actuators convert an electrical signal into a corresponding physical quantity such as movement, force, sound etc. An actuator is also classed as a transducer because it changes one type of physical quantity into another and is usually activated or operated by a low voltage command signal. Actuators can be classed as either binary or continuous devices based upon the number of stable states their output has.

For example, a relay is a binary actuator as it has two stable states, either energized and latched or de-energised and unlatched, while a motor is a continuous actuator because it can rotate through a full 360o motion.

Let’s explore some of the basic actuators you may use in your IoT projects –

### Servo Motors:



A Servo is a small device that incorporates a two wire DC motor, a gear train, a potentiometer, an integrated circuit, and a shaft (output spine). The shaft can be positioned to specific angular positions by sending the servo a coded signal.  Of the three wires that stick out from the servo casing, one is for power, one is for ground, and one is a control input line.

When a control signal is applied to a Servo that represents a desired output position of the servo shaft, it (servo) applies power to its DC motor until its shaft turns to that position. It uses the position-sensing device to determine the rotational position of the shaft, so it knows which way the motor must turn to move the shaft to the commanded position.

### 2. Stepper Motors:



Stepper motors are DC motors that move in discrete steps. They have multiple coils that are organized in groups called “phases”. By energizing each phase in sequence, the motor will rotate, one step at a time. With a computer controlled stepping, you can achieve very precise positioning and/or speed control.

A servomotor consumes power as it rotates to the commanded position but then the servomotor rests. Stepper motors continue to consume power to lock in and hold the commanded position.

### 3. DC Motors (Continuous Rotation Motors):

### DCMotor

Direct Current (DC) motor is the most common actuator used in electronics projects. They are simple, cheap, and easy to use. Also, they come in a great variety of sizes, to accommodate different tasks.  DC motors convert electrical into mechanical energy. They consist of permanent magnets and loops of wire inside. When current is applied, the wire loops generate a magnetic field, which reacts against the outside field of the static magnets.

### 4. Linear actuator:



A linear actuator is an actuator that creates motion in a straight line, in contrast to the circular motion of a conventional electric motor. Linear actuators are used in machine tools and industrial machinery, in computer peripherals such as disk drives and printers, in valves and dampers, and in many other places where linear motion is required.

### 5. Relay:



A relay is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. The advantage of relays is that it takes a relatively small amount of power to operate the relay coil, but the relay itself can be used to control motors, heaters, lamps or AC circuits which themselves can draw a lot more electrical power.

### 6. Solenoid:

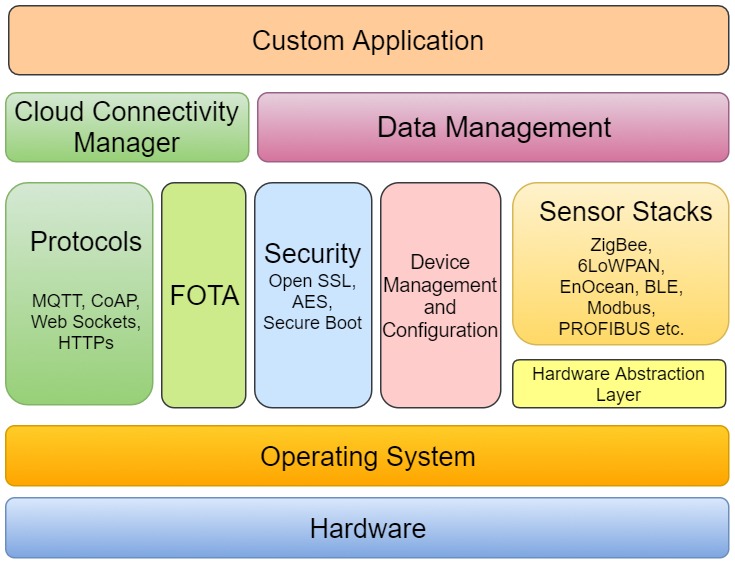


A solenoid is simply a specially designed electromagnet. Solenoids are inexpensive, and their use is primarily limited to on-off applications such as latching, locking, and triggering. They are frequently used in home appliances (e.g. washing machine valves), office equipment (e.g. copy machines), automobiles (e.g. door latches and the starter solenoid), pinball machines (e.g., plungers and bumpers), and factory automation.

GATEWAY

An Internet of Things (IoT) gateway is a physical device or software program that serves as the connection point between the cloud and controllers, sensors and intelligent devices. All data moving to the cloud, or vice versa, goes through the gateway, which can be either a dedicated hardware appliance or software program. An IoT gateway may also be referred to as an intelligent gateway or a control tier.

An IoT gateway aggregates sensor data, translates between sensor protocols, processes sensor data before sending it onward and more.



IOT gateway architecture

IOT GATEWAY HARDWARE

IoT Gateway Hardware comprises of processor/microcontroller, IoT sensors, protection circuitry, connectivity modules (e.g Zigbee, Bluetooth, WiFi and more).

Type of hardware (processor/microcontroller), processing speed and memory space is decided based on the Operating System of the IoT Gateway device.

A small to a medium scale application can run on a microcontroller; however if the gateway is expected to do complex operations a processor is needed.

## OPERATING SYSTEM

Selection of the Operating system is also largely dependent on the IoT application. If the gateway is to be designed for a simple to medium scale  application then a RTOS (Real Time Operating System) is used; however if the gateway has to perform considerably complex operations then Linux is preferred.

## HAL (Hardware Abstraction Layer)

Hardware Abstraction Layer supports reusability and portability of the IoT software. This layer makes the software design independent of the underlying hardware platform. Hence it helps to reduce the time and cost required to port the developed software application into a different hardware platform (during migration from the existing platform or re-design of the product line).

### IoT Sensors Stack

This layer basically consists of software stacks that serve as interfaces with IoT sensors modules. Specific stack(s) is/are integrated depending on the sensor interface the IoT Gateway has to support. Some of the populary integrated stacks are ZigBee, 6LoWPAN, EnOcean, BLE, Modbus, PROFIBUS and more.

### Device Management and Configuration

An IoT gateway needs to interface with different types of Sensor devices and each sensor node (used for capturing distinct data) has different set of properties. IoT Gateway device is required to keep track of all the connected devices/sensors. In addition to this, all the devices/sensors management and configuration tasks are performed at the IoT Gateway. Thus it is important that the IoT Gateway Device is easily configurable to manage IoT Sensor settings, properties and access control. The configuration and settings of all the IoT Sensor Devices is stored in the gateway device memory. This ensures that the last saved settings are available after every re-boot.

### Security

Gateway security is one of the key considerations in IoT gateway architecture during the IoT Gateway design process. The designed IoT gateway should ensure robust data security, device security and network security. Device security and device identity is implemented in the gateway hardware using Crypto Authentication chips.  To add further security to the IoT gateway hardware tampering is implemented. Secure boot is also implemented to ensure that the gateway doesn’t boot from an unauthorized firmware. All messages between gateway and cloud, and messages between Gateway sensor node is encrypted to ensure data integrity, and confidentiality of sensor nodes. Data to and from every node in a IoT application is encrypted to ensure network security.

### FOTA

Ensuring IoT Gateway security requires continuous and timely efforts; as an IoT Development Partner, one needs  to keep fixing the security loop holes fixed and maintain device integrity. FOTA updates ensure that the IoT Gateway software is updated with latest versions of security patches, OS, Firewalls and more. Within the IoT network, the gateway device periodically checks for firmware updates in the cloud and fetches the update. In case of failure IoT Gateway reverts to the last best known state. Before the update process begins, IoT Gateway checks if the available firmware version is from a trusted source.

### Data Communication Protocols

The IoT Gateway connects with the Cloud over Ethernet, Wi-Fi or a 4G/3G modem. Two way communication channel is established with the Cloud for data exchange and command(s) transfer. The underlying communication layer is UDP or TCP IP protocol. For ease of development and to maintain standardization, protocols like MQTT, CoAP, XMPP, AMQP are utilised. This is because handling and maintaining communication with cloud over raw socket is more complex process. Protocol(s) is/are selected considering the amount and frequency of the data that has to be shared with the Cloud.

### Data Management

Data Management includes data streaming, data filtering and data storing (in case of loss of connectivity with the cloud). IoT Gateway manages the data from sensor nodes to gateway and also the data from gateway to cloud. The challenge here is to minimize the delay to ensure data fidelity.

### Cloud Connectivity Manager

This layer is responsible for seamless connectivity with the cloud and also handles scenarios like reconnection, device state, heartbeat message, and gateway device authentication with the cloud.

### Custom Application

IoT Gateway application is custom designed as per the business needs. Gateway application interacts with services and functions from all the other layers or modules to manage data between sensor node and gateway and from gateway to cloud in an efficient, secure and responsive manner.

### Gateway Data Transfer

IoT gateway can be connected to the internet for data transfer using Ethernet, 4G/3G/GPRS modem or Wifi. Non-GPRS network is the most preferred mode of internet connectivity. This is due to the cost effectiveness of the data transfer through Wifi or Ethernet.

The gateway should have in-built intelligence to analyze and decide which data should be transferred over the network for processing and which data can be cached for offline processing to save the data transfer cost and processing power of the main application.

After understanding the architecture, one realizes that the Design and development of an IoT Gateway device is a work of art!

As an IoT software and hardware developer, it is very important to understand the business needs (and logic) of the IoT Application.

This understanding of the IoT Application is an important factor that contributes to the development of a win-win IoT Gateway design

Though some very popular off-the shelf IoT Gateway solutions are available, but certain customization becomes a necessity to transform a product concept into business reality!

Prototype

In hardware design, a prototype is a "hand-built" model that represents a manufactured (easily replicable) product sufficiently for designers to visualize and test the design.

Protocol

A **protocol** is a standard set of rules that allow electronic devices to communicate with each other. These rules include what type of data may be transmitted, what commands are used to send and receive data, and how data transfers are confirmed.

UTF

Stands for Stands for "[Unicode](https://techterms.com/definition/unicode) Transformation Format." UTF refers to several types of Unicode character encodings, including UTF-7, UTF-8, UTF-16, and UTF-32

 **UTF-7** - uses 7 bits for each character. It was designed to represent ASCII characters in email messages that required Unicode encoding.

 **UTF-8** - the most popular type of Unicode encoding. It uses one byte for standard English letters and symbols, two bytes for additional Latin and Middle Eastern characters, and three bytes for Asian characters. Additional characters can be represented using four bytes. UTF-8 is backwards compatible with ASCII, since the first 128 characters are mapped to the same values.

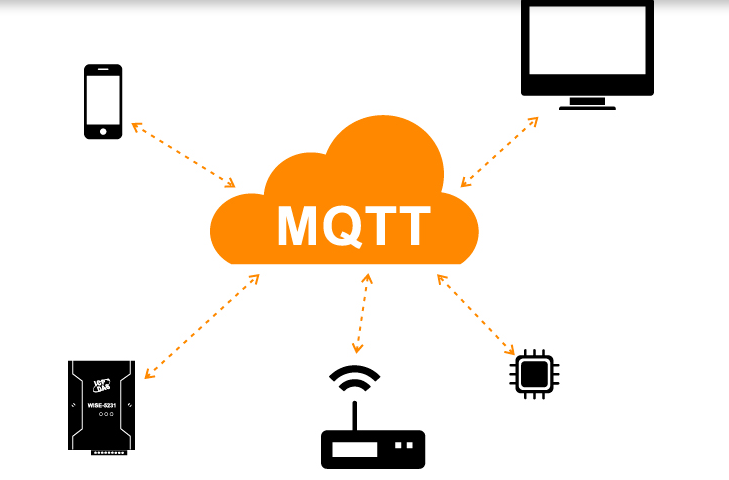
Difference between http and https

Hyper Text Transfer Protocol Secure (**HTTPS**) is the secure version of **HTTP**, the protocol over which data is sent between your browser and the website that you are connected to. The 'S' at the end of **HTTPS** stands for 'Secure'. It means all communications between your browser and the website are encrypted.

MQTT protocol

MQTT is one of the most commonly used protocols in IoT projects. It stands for Message Queuing Telemetry Transport.

In addition, it is designed as a lightweight messaging protocol that uses publish/subscribe operations to exchange data between clients and the server. Furthermore, its small size, low power usage, minimized data packets and ease of implementation make the protocol ideal of the “machine-to-machine” or “Internet of Things” world.



# Why MQTT?

MQTT has unique features you can hardly find in other protocols, like:

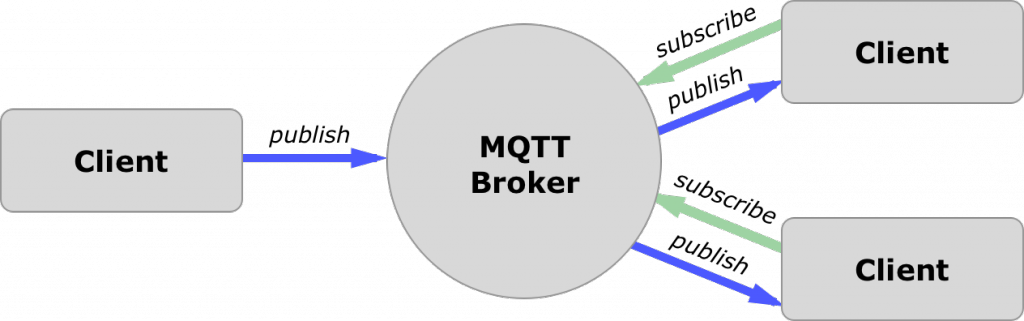
* It’s a lightweight protocol. So, it’s easy to implement in software and fast in data transmission.
* It’s based on a messaging technique. Of course, you know how fast your messenger/WhatsApp message delivery is. Likewise, the MQTT protocol.
* Minimized data packets. Hence, low network usage.
* Low power usage. As a result, it saves the connected device’s battery.
* It’s real time! That’s is specifically what makes it perfect for IoT applications.

# How MQTT works

Like any other internet protocol, MQTT is based on clients and a server. Likewise, the server is the guy who is responsible for handling the client’s requests of receiving or sending data between each other.

MQTT server is called a broker and the clients are simply the connected devices.  
So:

* When a device (a client) wants to send data to the broker, we call this operation a “publish”.
* When a device (a client) wants to receive data from the broker, we call this operation a “subscribe”.



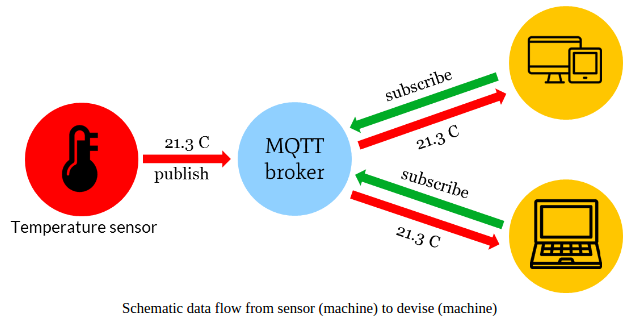
In addition, These clients are publishing and subscribing to topics. So, the broker here is the one that handles the publishing/subscribing actions to the target topics.

# Example:

Let’s say there is a device that has a temperature sensor. Certainly, it wants to send his readings to the broker. On the other side, a phone/desktop application wants to receive this temperature value. Therefore, 2 things will happen:

* The device defines the topic it wants to publish on, ex: “temp”. Then, it publishes the message “temperature value”.
* The phone/desktop application subscribes to the topic “temp”. Then, it receives the message that the device has published, which is the temperature value.

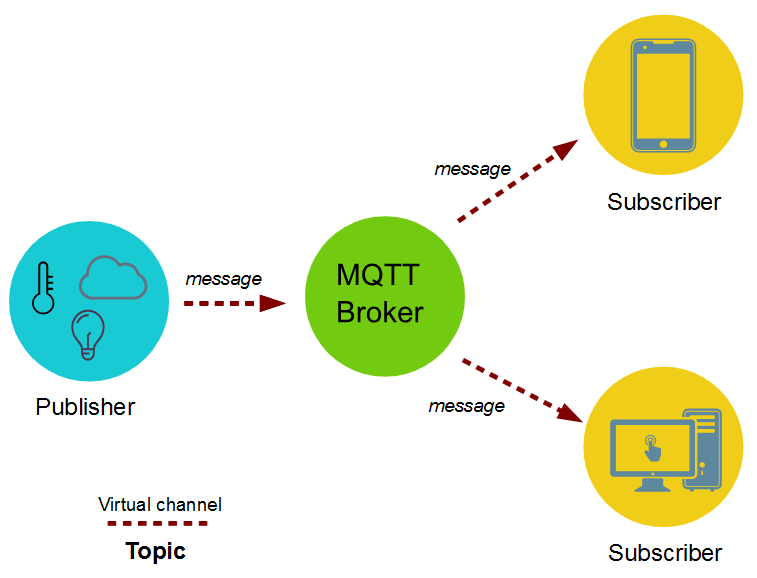
Again, the broker role here is to take the message “temperature value” and deliver it to phone/desktop application.



# MQTT Components:

That takes us to the MQTT components, which are 5 as follows:

* Broker, which is the server that handles the data transmission between the clients.
* A topic, which is the place a device want to put or retrieve a message to/from.
* The message, which is the data that a device receives “when subscribing” from a topic or send “when publishing” to a topic.
* Publish, is the process a device does to send its message to the broker.
* Subscribe, where a device does to retrieve a message from the broker.



# How many devices you can connect to a broker

The number of connected devices “clients” to the broker depends on the broker service provider.

In fact, it can reach a massive number of clients those are publishing and subscribing all the time.

but the amazing part of this isn’t only the huge number of these connected devices but also any the fact that any device can get any other device’s data at any time. As a result, the applications based on these quickly shared data are limitless.

# Why not HTTP

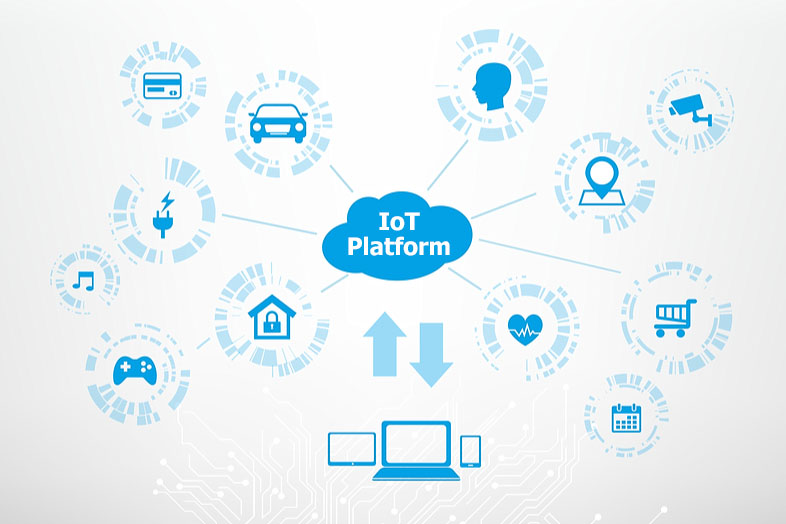
HTTP is slower, more overhead and power consuming protocol than MQTT. So, let’s get into each one separately:

* Slower: because it uses bigger data packets to communicate with the server.
* Overhead: HTTP request opens and closes the connection at each request, while MQTT stays online to make the channel always open between the broker “server” and clients.
* Power consuming: since it takes a longer time and more data packets, therefore it uses much power.  
  

# Broker in the IoT system components

In the previous [IoT](https://1sheeld.com/iot-understanding-concepts/) blog, we discussed the IoT system in details and its 4 components. I would recommend you go through it if this is your first time reading about IoT.

One of these components was the IoT platform. It’s responsible for connecting the devices together, indeed.

Though, the IoT platform is just a cloud-based software that implements a broker server and some software with GUI to allow user control and monitor his connected devices.  
  
  
Therefore, MQTT broker is included implicitly inside the IoT platform.

But we won’t use an IoT platform in our upcoming IoT blogs and tutorials.

That’s because we will only connect a few devices for small projects. So, IoT platform isn’t needed here. But it was necessary to get you a bit knowledge about it.

# Which broker to use?

There are many brokers that implement the MQTT protocol. One of the most popular and commonly used is the mosquito broker.

# Mosquitto broker https://1sheeld.com/wp-content/uploads/2018/07/mosquitto_network-287x300.png

Mosquitto is an open source message broker that implements the MQTT protocol. It’s lightweight and suitable for use on all devices from a low power single board like Arduino, ESP8266 to full computers and servers.

But rather than using the Mosquitto on a local PC, you will need to use a cloud-based server that implements the Mosquitto broker. That’s necessary to make your IoT projects controllable over the internet.

Cloud-based Mosquitto brokers are many, like:

* ThingMQ
* ThingStudio
* MQTT.io
* Heroku
* CloudMQTT

And I will focus on the last one in this and the upcoming blogs.

**CloudMQTT broker**

CloudMQTT is one of the best and easiest cloud-based Mosquitto broker.

CloudMQTT has a free plan that allows you to set up your own CloudMQTT broker instance that will run on their hardware servers. Hence, you can have an online broker that’s ready to use in your IoT project.

It also has a well designed GUI to monitor the publishing and subscribing processes and topics through an easy to use WebSocket UI.

Zigbee

### ****What is ZigBee Protocol?****

The ZigBee wireless technology is basically a openly available global standard to address the uniques needs of low-power, low-cost wireless M2M(machine-to-machine) networks and also Internet-of-Things(IoT). It operates on IEEE 802.15.4 physical radio specification and operates even in unlicensed band including 2.4 GHz, 900 MHz and 868 MHz.

In 2003, the 802.15.4 specification on which the ZigBee stack operates got an official approval and recognition from the [Institute of Electrical and Electronics Engineers](http://www.ieee.org/portal/site) (IEEE). This specification is a packet-based radio protocol that is intended for the low-cost, battery-operated devices. This protocol offers devices to have battery life lasting for years and also allows them to communicate in a variety of network topologies.

### ****ZigBee Protocol: Basics****

The ZigBee protocol is robust in a way to be used in various hostile RF, Wi-Fi, Bluetooth based environments, which are common in various industrial applications. ZigBee protocol features a unique channel agility mechanism and takes full advantage of the IEEE 802.15.4 proven interference avoidance techniques. Some of the technical details for the ZigBee protocol are listed below:

* The ZigBee protocol operates globally on a single frequency of 2.4 GHz.
* ZigBee offers wireless range of 70m indoors and and 400m outdoors.
* It offers networking flexibility to covers homes of all size by offering support for multiple networks like point-to-point, point-to-multipoint mesh-networks.
* Low latency and Low Duty cycle leads to lower power consumption giving sensors the long-lasting battery life for up to 7 years.
* Direct Sequence Spread Spectrum (DSSS) helps to offer higher data rate for quicker responses.
* Caters to thousands of devices for spread networks.
* ZigBee uses AES 128 encryption (government, commercial and military grade encryption used across the Internet) thus protecting your information over the air transfers.
* ZigBee can easily integrate monitoring and control of lights systems, security systems, convenience and motion detection.
* The mesh-network operability of ZigBee reduces the chances of failure at nodes and the ad-hoc routing offers greater stability.

### ****Applications of ZigBee Technology****

Because of its three major USPs of being low-cost, low-power consumption and having faster wireless connectivity, the ZigBee protocol caters to a lot of applications like industrial automation, home automation, smart metering, smart grids etc. Also with it low-power requirements, it ensures seamless operation of various sensor equipments offering years of battery-life. Here are some of the areas where ZigBee is widely used.

* **Industrial Automation:** ZigBee offers a faster and low-cost communication that can communicate with almost all devices in factories and centralise them at one place making it easy for you to monitor every process and thereby optimise the control process.ZigBee protocol also finds its presence in many medical and scientific equipments such as personal chronic monitoring, sports and fitness trackers, and can even be used for remote patient monitoring.
* **Smart Metering and Smart Grid Monitoring:**In case of smart metering, ZigBee is used for better energy consumption response, security over power theft, pricing support etc. Additionally in case of smart grids, ZigBee is even used for reactive power management, fault locations, remote temperature monitoring, etc.
* **Home Automation:**ZigBee is one of the most widely used protocol in most of the home automation equipments. Right from offering lighting system solutions, sensor responsive solutions to security solutions and surveillance, ZigBee has its presence everywhere.

### ****ZigBee for Home Automation****



Source: ZigBee Alliance

ZigBee protocol is widely used for home automation solutions and caters to complete holistic solutions of lighting control, security control, comfort control and even energy management.

There are several well-known globals brands in home automation that use ZigBee for their devices. Since Zigbee is cross-compatible and interoperable, it makes managing multi-vendor devices easy and simple. If a device is ZigBee Home Automation (HA 2.1) compliant, you can be rest assured that it will work with your automation system, irrespective of the vendor.

The mesh-routing network of ZigBee wherein one device can talk to multiple device and data packets travel on no fixed routes, offers better flexibility and faster communication across devices.

Some of the features of ZigBee for Home Automation include:

* Simplified setup and maintenance
* Ideal for new construction and remodelling
* ZigBee gives access to devices anywhere from the world just from your smartphone
* Monitors power use and allows you to turn on/off devices from remote locations
* Built in security with interference avoidance techniques ensures better/enhanced security and worry-free operations.
* Help you customise lighting scenes based on daily schedules, events and activities.
* Due to low-power consumption of the ZigBee protocol, your security sensors can work for a period of 7 years.

### ****Technical Details****

#### ****Hardware****

ZigBee devices are categorised into following three kinds:

* 1. **ZigBee Co-ordinator (ZC):**This is the most important device as it forms the root of the network tree and helps to bridge to other networks. This means that you will find one ZigBee coordinator in each network as this is the device responsible for the start of the network. This device contains all the information of the network and functions as a Trust Center & repository for the security keys.
  2. **ZigBee Router (ZR):**In addition to running an application function, its is used to route the data from other devices and help it reach the destination.
  3. **ZigBee End-Device (ZED):**The end device contains just enough functionality to talk to either the co-ordinator or the router. Note that it cannot rely data from other devices. This causes the node to stay asleep for a significant time thereby increasing battery life to a considerable extent. A **ZED**device requires least amount of energy as compared to the **ZC or ZR.**