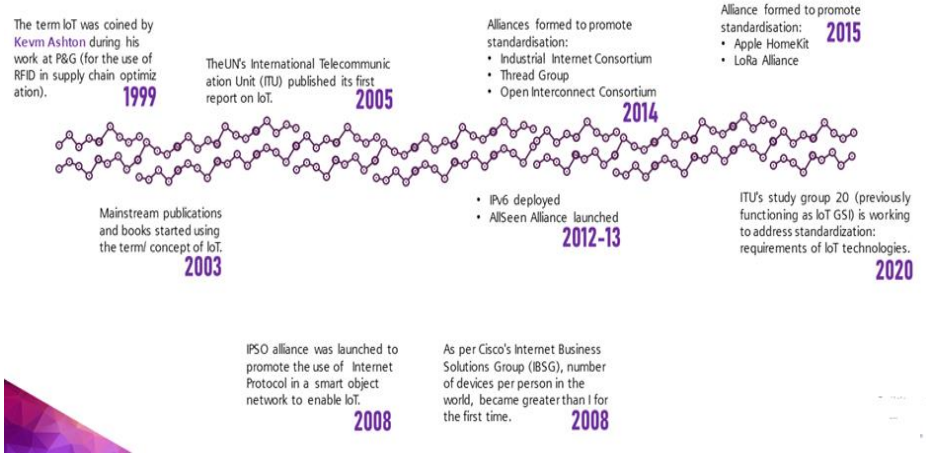
Module-II

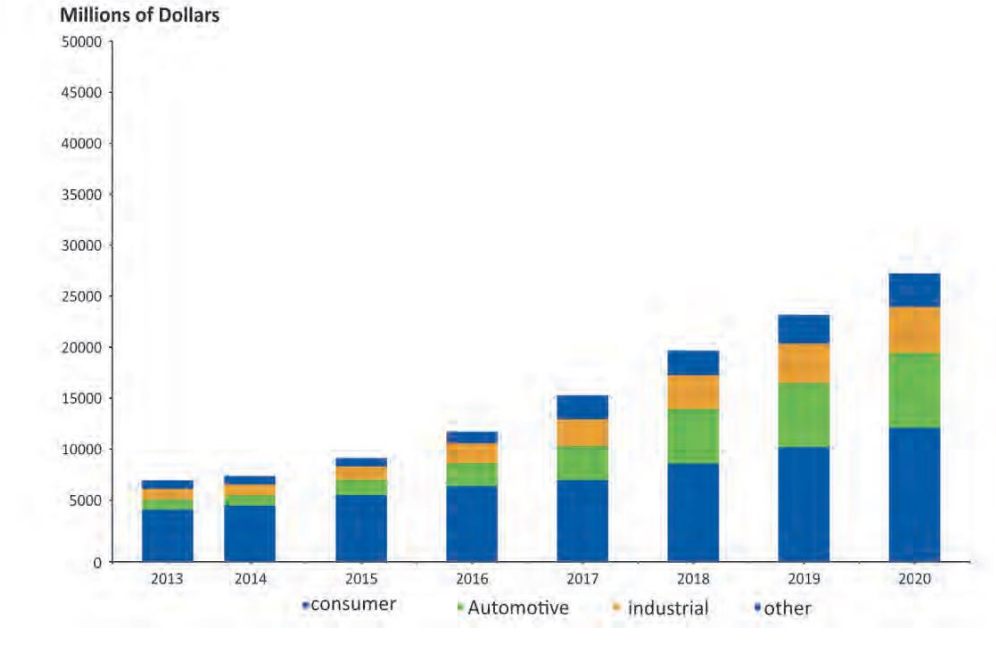
Evolution of IoT and the trends, Impact of IoT on businesses and society, Existing IoT use cases and applications across industries.

History of IoT

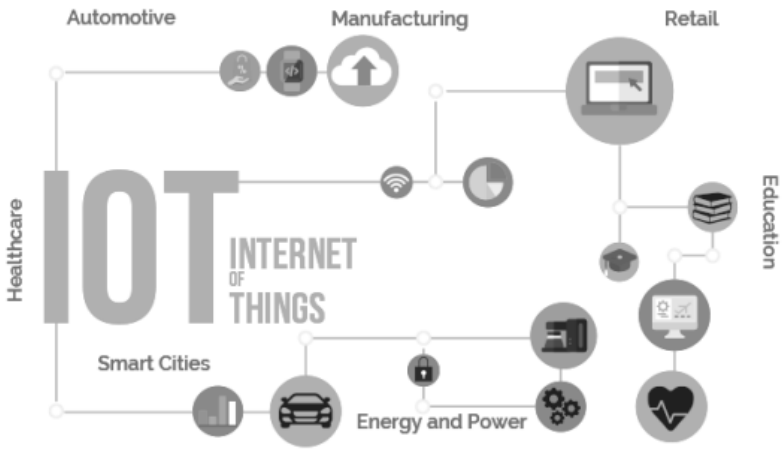
Evolution of IoT



Revolution of IoT



IoT Applications



Sectors & Businesses Impacted by IoT

- Manufacturing
- Financial Services
- Communication
- Education
- Business
- Transport
- Medical - Health

Characteristics of IoT

Low energy requirement

Scalable solution

The solution operating under very secured environment

Interoperability between different communication technologies

Open API/Open source

Characteristics of IoT

- Dynamic & Self-Adapting
- Self –Configuring
- Interoperable communication Protocols
- Unique Identity
- Integrated into Information Network

- **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, user's context, or sensed environment.**
- For example, consider a **surveillance system** comprising of a number of surveillance cameras.
- The surveillance cameras can adapt their modes (to normal or infra-red night modes) based on whether it is day or night.
- Cameras could switch from lower resolution to higher resolution modes when any motion is detected and alert nearby cameras to do the same.
- The surveillance system is adapting itself based on the **context and changing (e.g., dynamic) conditions.**

- **Self-Configuring:** IoT devices may have self-configuring capability,
- allowing a large number of devices to work together to provide certain functionality.
- These devices have the ability
  - **to configure themselves** (in association with the IoT infrastructure),
  - **setup the networking,**
  - **and fetch latest software upgrades with minimal manual or user intervention.**

- **Interoperable Communication Protocols:**
  - IoT devices support a number of interoperable communication protocols and
  - can communicate with other devices and also with the infrastructure.

- **Unique Identity:**
  - Each IoT device has a unique identity and a unique identifier (such as an IP address or a URI).
  - IoT systems may have **intelligent interfaces** which adapt based on the **context**, allow communicating with users and the environmental contexts.
- **IoT device interfaces allow users**
  - **to query the devices,**
  - **monitor their status, and**
  - **control them remotely, in association with the control, configuration and management infrastructure.**

- **Integrated into Information Network:**
  - IoT devices are usually integrated into the **information network**
  - **that allows them to communicate and exchange data with other devices and systems.**
- IoT devices can be
  - dynamically discovered in the network,
  - by other devices and/or the network, and
  - have the capability to describe themselves to other devices or user applications.

For example:

- a weather monitoring node can describe its monitoring capabilities to another connected node.
- **Integration into the information network helps in making IoT systems "smarter"**
- **due to the collective intelligence of the individual devices in collaboration with the infrastructure.**
- The data from a large number of connected weather monitoring IoT nodes can be aggregated and analyzed to predict the weather.

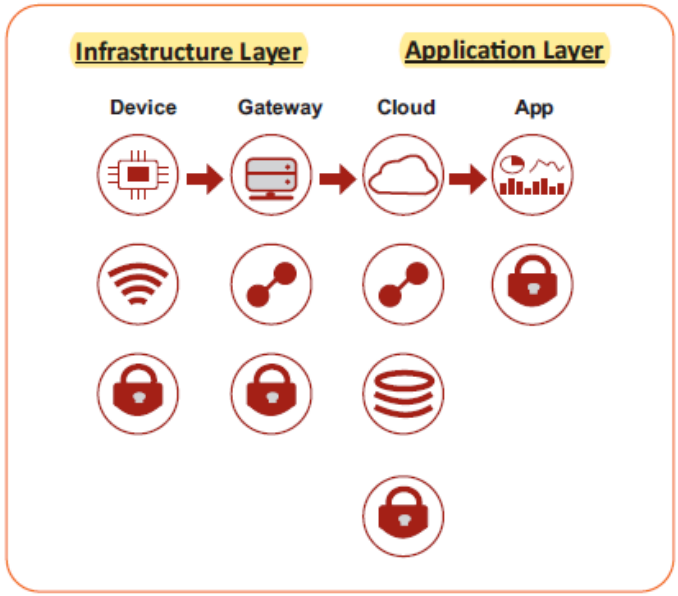
IoT Ecosystem things

- Device
  - An IoT device allows identification, remote sensing, actuating and remote monitoring capabilities.
- Resource
  - Resources are software components on the IoT device for accessing, processing, and storing sensor information, or controlling actuators connected to the device. Resources also include the software components that enable network access for the device.
- Internet or Controller service
  - Controller service is a native service that runs on the device and interacts with the web services. Controller service sends data from the device to the web service and receives commands from the application (via web services) for controlling the device.

IoT Ecosystem things

- Data base
  - Database can be either local or in the cloud and stores the data generated by the IoT device.
- Cloud (Web Service)
  - Web services serve as a link between the IoT device, application, database and analysis components. Web service can be either implemented using HTTP and REST principles (REST service) or using WebSocket protocol (WebSocket service).
- Applications (Analysis component)
  - The Analysis Component is responsible for analyzing the IoT data and generate results in a form which are easy for the user to understand.
- Platforms & Products
  - IoT applications provide an **interface** that the users can use to control and monitor various aspects of the IoT system. Applications also allow users to view the system status and view the processed data.

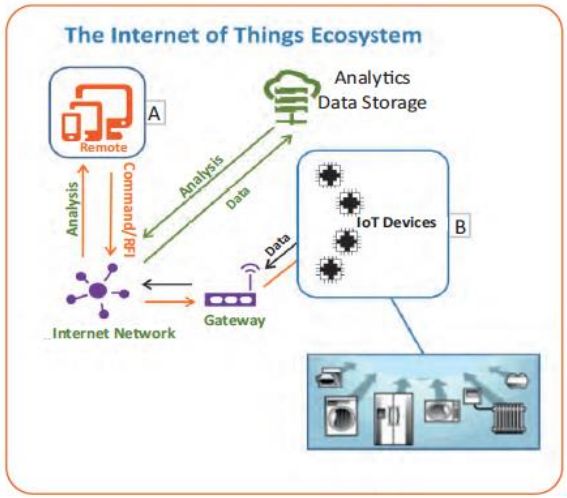
Working Principle and mechanism of an IoT System



Contd.,

- Devices consist of Sensors & Sensor Technology. They collect a wide variety of data ranging from temperature, pressure, location, weather/environment conditions, grid parameters, health essentials of a patient, etc.
- Gateways as the name suggests, are the gateways to the internet for all the devices or things that need to interact with it. They help to connect the sensor nodes in the internal network with the external Internet, by collecting data from the sensor nodes and further transmitting it to the internet infrastructure.
- The data transmitted through the gateway is then stored and processed securely within the Cloud infrastructure using the Big Data analytics engine. The data thus processed performs intelligent actions. This is what makes the 'Smart Devices'!
- The Applications help the end users to control and monitor their devices from remote locations. They not only send important information on the hand-held devices or PCs but also help to send commands back to the Smart Devices.

IoT ecosystem (characteristic general arrangement of IoT)-based on Home based application



It is basically control the loads of the AC, refrigerator, washing machine, etc. through the Internet network remotely on the basic principle of the Internet of Things

Working principle

- The devices (See B in the Figure) should consist of power management modules and status data report module interfaced to RF modules used for communication. The status of the device will also be communicated and displayed on the web page.
- The devices to be monitored/controlled communicate through RF modules in their signal processing for WiFi / ZigBee/ Bluetooth/ radio transceiver (RFID) etc, or through wired LAN.
- The data for control / monitor is harnessed through internet network to a remote control / monitoring center. PC system, laptop, tablet A or smart phone is used for remote control centers and monitoring. (See A in the figure).

Contd.,

- A Wi-Fi Module (Gateway)is configured with any nearby wireless modem to access internet. The received Internet commands are fed to the Wi-Fi module.
- The program within the Wi-Fi Module executes the commands received based on which the device gets activated through Bluetooth or ZigBee module interfaced to Wi-Fi Module.
- This Gateway will in turn communicate the data to the cloud for analytics and remote access.
- A real-time webpage will be essential with a user configurable front end to control and monitor (through Mobile phone, Tablet and PC) the load·
  - The data sent from a password protected webpage returns commands through allotted IP fed to it.

Sensors, Actuators, wireless sensor networks, Gateways, cloud support, Big data analysis

TECHNOLOGIES OR COMPONENTS IN IOT

Things

- **Things** can be a piece of metal machinery that a device is attached to.
- It considers the non-electronic ,non- people item.
  - Sensors and actuators that collects the physical world context

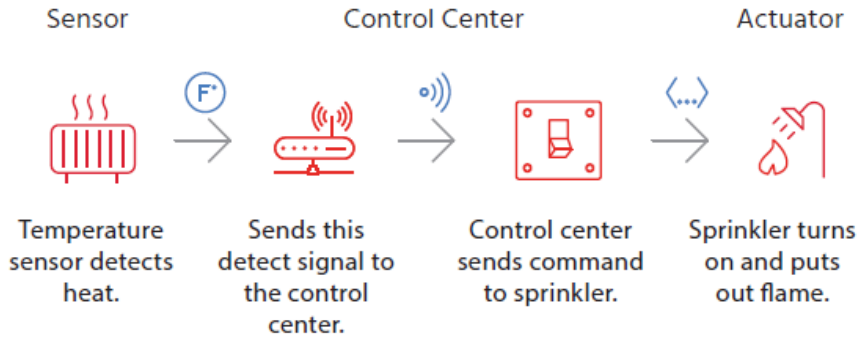
IoT Hardware

- IoT hardware platform takes charge of collecting, storing, and processing data based on the connection of the Internet
- IoT Hardware Components
  - Sensors
  - Actuators
  - Processors & Controllers

Sensors & Actuators

- Sensor and actuators are kind of transducers.
- A transducer & sensor
  - A physical device
  - converts one form of energy into another.
  - the transducer converts some physical phenomenon into an electrical impulse that can then be interpreted to determine a reading
    - Eg:A microphone is a sensor that takes vibrational energy (sound waves), and converts it to electrical energy in a useful way for other components in the system to correlate back to the original sound .
- An Actuator
  - Does the operation in reverse.
  - It takes an electrical input and turns it into physical action
  - Eg: an electric motor, a hydraulic system, and an alarm are all different types of actuators
- IoT systems, a sensor may collect information and route to a control center where a decision is made and a corresponding command is sent back to an actuator in response to that sensed input
- **Processor & Microcontroller**
  - Connects sensor and actuator to the Internet
  - Operates corresponding instructions

Sensor to Actuator Flow for smart kitchen



IoT Hardware (Various sensors)

- Mobile phone based sensors
  - Motion sensors, Accelerometers, Magnetometer and compass, Environmental sensors like thermometers, barometers (pressure sensors), Cameras Microphone
- General Environmental and chemical sensors
  - Temperature, Heat, Pressure, Moisture, Humidity, Proximity, Motion, vibration.
  - Can measure water pollution and air pollution parameters
- Wearable sensing devices
  - Fitbit watches,
  - monitoring tattoos patches (vital param
  - Brain Signal Sensing Band
- RFID
  - Transmit pre-embedded 'information' directly to the RFID Reader

IoT Hardware

❖ **Sensors**

- Sensor Types

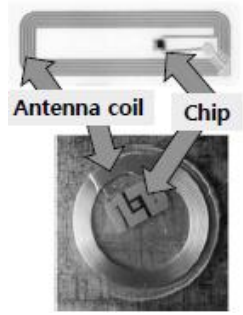
Type	Detect	Model	Measurement	Shape
Temperature/ Humidity sensor	Actual Temp. and humidity	DHT11, DHT22	Temperature: -40 ~ 80 ℃ Humidity: 0 ~ 100% RH	
Pressure sensor	Pressure w. r. t. atmospheric pressure	SPD005G SPD100G	SPD005G: 0 kPa ~ 35 kPa SPD100G: 0 kPa ~ 650 kPa	
Flow sensor	Rate of fluid flow	YF-S201	1 to 30 Liters/Minute	
Imaging sensor	Conversion of variable attenuation of image into signal	OV7670	Maximum 30 fps, 640 x 480 VGA resolutions (= 0.3 Megapixels)	
Ultrasonic sensor	Presence of an object by ultrasonic wave	HC-SR04	2 ~ 400 cm non-contact measurement @ 40 Hz	

IoT Hardware : Actuators

- **Actuator types**
  - Electrical actuator
    - Converts energy to mechanical torque
  - Mechanical motion
    - Motion Types: Linear, Rotary, Oscillatory, etc
      - Mechanical linear actuator
        - » Converts rotary motion to linear motion
  - Hydraulic / Pneumatic actuator
    - Convert fluidal (liquid / gas) compression to a determined work.
      - The function of the actuator in hydraulic control systems is **to convert the hydraulic energy supplied by the pump and processed by the control elements into useful work**.

RFID (Radio Frequency Identification)

- RFID chip (tag) holds information about a 'thing'
- RFID chip is attached and transfers data to the reader
- Antenna is used to receive energy from the Reader that is used to operate the RFID device
- RFID tag transmits its information back to the reader



RFID Types

Type	Working frequency	Read range	Standard
Low frequency RFID	125 ~134.3 kHz	10~ 30 cm	ISO 14223 ISO/IEC 18000-2
High frequency RFID	13.56 MHz	10 cm ~ 1 m	ISO 15693 ECMA-340, ISO/IEC 18092 NFC (Near Field Communication)
Ultra-high Frequency RFID	860 ~ 960 MHz	12 m	ISO 18000-63

RFID (Radio Frequency Identification)

- RFID enables efficient management, tracking, and monitoring processes
- Logistics and supply chain applications
- RFID R&D
  - Streams of data support
  - Chip design
  - Energy usage optimization
  - Automatic meter reading
  - Home automation applications
  - Vehicle & transportation applications

Processors & Microcontrollers

- IoT Device Platform
  - Processors & Microcontrollers
  - HW, SW, Sensors, and Interfaces
  - Networking Modules (WAN, LAN, WLAN, PAN)
- Provides HW interfaces (USB, GPIO, UART etc.) to connect User →Sensor → Actuator
- OS (Operating System) supports a SW interface to control HW resources
  - Power, Memory, File I/O, etc.

Arduino

- Open-source microcontroller & hardware
  - May also refer to an 'Open-source Arduino hardware and software project'
- Single-board microcontrollers and kits
  - Enables easy sensing and controlling objects
  - Popular for IoT development
- Arduino board circuit design and the IDE (Integrated Development Environment) are available on the Arduino website
- User-specific programs can be developed and uploaded using the IDE
  - Uses USB connection to an Arduino board

Arduino Product Types

- Arduino Uno R3 (Entry and General purpose)

- Arduino Yun (IoT)

- Arduino Lilypad (Wearable)


Author: leah buechley
https://commons.wikimedia.org/wiki/File:Flexible_Lilypad_Arduino.jpg

Atmel

IoT Hardware
• megaAVR (ATmega) series are adopted as a **Arduino** physical computing platform

Arduino board	Description	Atmel AVR
Arduino Uno	Entry level	ATmega328P
Arduino Leonardo		ATmega32u4
Arduino Yun	IoT	ATmega32u4
Arduino Ethernet		ATmega328
Arduino Lilypad (USB, Mainboard, Simple)	Wearable	ATmega32u4 ATmega168 ATmega328

Raspberry Pi

- Developed by Raspberry Pi Foundation in the UK
- Developed as a low cost single-board computer to promote basic computer science skills in schools
- Supports general computations and basic web server functions
- Specifications
  - HW: Broadcom SoC, ARM CPU, On-chip GPU
  - SW: Raspbian OS

Raspberry Pi Product Series

- Mainline: Raspberry Pi 3 Model B

  - 1.2 GHz ARM Cortex CPU based micro computer for general IoT functionality
- Subline: Raspberry Pi Zero W

  - Smaller size and restricted I/O, GPIO capabilities

BeagleBoard

- Open-source single-board computer produced by Texas Instruments
- Fully functional basic computer
- Supports various OSs
  - Linux, Android
- Includes advanced features
- Little more expensive than other single-board computers

BeagleBoard

Beaglebone black

- Key features
  - Very low power requirements (~2W)
  - PRU (Programmable Real-time Unit)
    - Used for deterministic latency (5ns per instruction) and delay-sensitive applications
    - Enhanced processor with image and 3D graphics processing

IoT Device Platforms

	Arduino Uno	Raspberry Pi 3 Model B	Beaglebone Black
Category	Microcontroller	Single-board micro computer	Single-board micro computer
SoC/CPU	16 MHz ATMega 328	• Broadcom BCM2837 SoC • 1.2 GHz ARM Cortex-A53 Quad-core @ 700 MHz	• Sitara AM3358 • 1 GHz ARM Cortex-A8 Single core @ 1000 MHz + Dual PRU @ 200 MHz
Memory	2 KB SRAM / 32 KB Flash	1 GB LPDDR2 / Micro SDHC support	512 MB DDR3 / 4GB Micro SDHC
I/O	14 (Digital GPIO) 6 (10-bit analog Input)	40 (Digital GPIO), 4 USB 2.0	69 GPIO 4 UART Serial, 8 PWM
Size	68.6 x 53.4 mm, 25 g	85.60 x 56.5 mm, 45 g	86.40 x 53.3 mm, 39.68 g

	Arduino Uno	Raspberry Pi 3 Model B	Beaglebone Black
Operating System	n/a, Arduino IDE for IDE	Linux (Raspbian)	Linux (Ubuntu, Devian, Android), Windows
Audio	n/a	3.5 mm analog, HDMI	Micro-HDMI
Video	n/a	HDMI	Micro-HDMI
Network	n/a, Extra shield required	Bluetooth 4.1 Classic, BLE 10/100 Mbps Ethernet 2.4 GHz Wi-Fi 802.11n	10 / 100 Mbps Ethernet
Price	$ 29.95	$ 35	$ 55

Embedded Systems in IoT

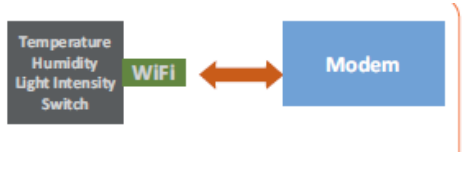
- Embedded hardware devices, micro controllers etc, are the ones that process the data.
  - The data from the sensors and actuators are usable and process them by implementing through the embedded system.
  - Embedding the hardware components through software.
  - Processors are the brain of the IoT system. Their main function is to process the data from the sensors according to an algorithm to extract the valuable information like status, parameter values, alerts etc., and control actuators in some cases in the system.
  - Processors work on a real-time basis. Changes in their operational limits can be set by applications depending on customer requirements.
  - These are also responsible for securing the data i.e. performing encryption and decryption of the data.

WSN (Wireless Sensor Networks)

- Efficient, low cost, low power devices for use in remote sensing applications
  - Low power integrated circuits and wireless communications
- A large number of intelligent sensors collect raw data, and create valuable services by processing, analysing, and spreading data
- Integrating data from sensors
  - Challenges are related to limited processing capability and storage, and sensor data sharing for multiple device/system cooperation

Wireless Sensor Networks

- With the rapid technological development of sensors, WSNs will become the key technology for IoT.
- **Wireless Sensor Network (WSN)** is a network formed by a large number of sensor nodes where each node is equipped with a sensor to detect physical phenomenon such as light, heat, pressure, etc.
- A sensor has the ability to capture anything from location to the device orientation.
- Collectively, these sensors produce a huge amount of data, both in unstructured form (such as picture or videos) as well as structured (such as GPS or acceleration data).
- These “devices” are perpetually connected to the Internet over WiFi, 3G or 4G.



IoT Cloud Computing Support

- For Advanced IoT services, IoT networks may need to collect, analyze, and process segments of raw data and turn it in into operational control information
- Advanced IoT services will need support of Cloud computing
  - Numerous IoT connections will be made to various devices and sensors
  - Many IoT devices will not have (PC or smartphone level) sufficient data processing capability or interoperability functionality

Cloud Computing

Contd.,

- A **cloud** refers to a distinct IT environment that is designed for the purpose of remotely provisioning **scalable and measured IT resources**
- cloud computing is a type of **Internet-based computing**, where different services such as servers, storage and applications are delivered to an organization's computers and devices through the Internet**.**
- it is a type of computing that relies on **sharing computing resources** rather than having local servers or personal devices to handle applications.
- It is a means of providing **remote access** to a set of decentralized IT resources.

Cloud Computing

- IoT applications will need support from a reliable, fast, and agile computing platform
- IoT devices can overcome lack of Software, Firmware, Memory Storage, Hardware, Data Processing capability through Cloud computing
- Cloud service models
  - SaaS (Software as a Service)
  - PaaS (Platform as a Service)
  - IaaS (Infrastructure as a Service)

Big Data Analytics

Contd.,

- Big data analytics is the process of **examining large and varied data sets i.e., big data to uncover hidden patterns, unknown correlations, market trends, customer preferences** and other useful information that can help organizations make more-informed business decisions.
- Analysis of the data coming from a device or other sources can be used to send information to a remote/local user regarding the device operation, or this analysis can be translated into commands sent to the device to influence its operation.
  - For example, the connected lamp can be turned on from the Cloud, the analysis shows that it is dark in the lamp's location and a shows that someone has entered the house.

Classification of IoT

- **Consumer IoT**, through home automation, infotainment or entertainment purpose and popular devices like wearables (fitness, lifestyle) etc. is driving volumes growth.
- **Industrial IoT**, through initiatives like smart manufacturing (Industry 4.0), smart cities, smart transportation, smart buildings, smart energy, etc. is driving revenue growth.

What kind of device ?Consumer/Industrial

- ALEXA, SIRI
- Super Market prediction system
- Wearable devices

Why we need these

- Different kinds of sensors
- Different kinds of actuators discussed
- What are actuator used in kitchen monitoring system
- Different kinds of controller / processors discussed
- Cloud component
- Various services of cloud platform
- Big data

Topics to be covered

- Physical design of IoT
  - Things
    - Block diagram of IoT
  - IoT protocols
- Logical design of IoT
  - IoT Functional Blocks
  - IoT Communication Models
  - IoT Communication APIs

Physical Design & Logical Design

- **Physical Design of IoT** system refers to **IoT Devices and IoT Protocols**.
- **Logical design of IoT** system refers to an abstract representation of the entities & processes without going into the low-level specifies of the implementation.

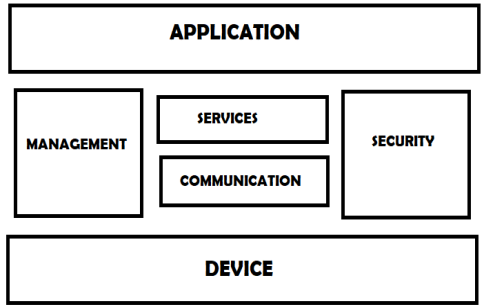
Logical design of IoT

Logical Design of IoT

- An abstract representation of the entities & processes without going into the low-level specifies of the implementation.
  - IoT Functional Blocks
  - IoT Communication Models
  - IoT Communication APIs

IoT Functional Blocks

- provide the system the capabilities for identification, sensing, actuation, communication and management.
  - Device
  - Communication
  - Services
  - Management
  - Security
  - Application



Functional Blocks

- **Device:** An IoT system comprises of devices that provide sensing, actuation, monitoring and control functions.
- **Communication:** Handles the communication for the IoT system.
- **Services:** services for **device monitoring, device control service, data publishing services** and services for device discovery.
- **Management:** this blocks provides various functions to govern the IoT system.
- **Security:** this block secures the IoT system and by providing functions such as authentication , authorization, message and content integrity, and data security.
- **Application:** This is an interface that the users can use to control and monitor various aspects of the IoT system. Application also allow users to view the system status and view or analyze the processed data.

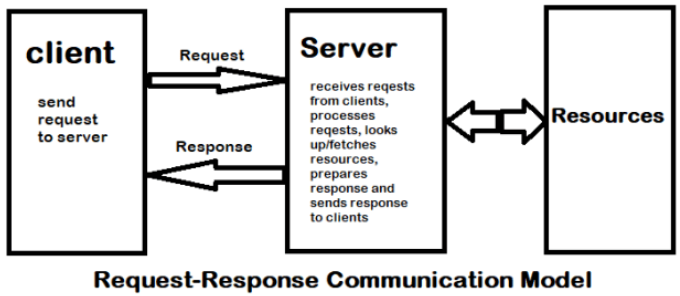
IoT Communication Models

- Request-Response Model
- Publish-Subscribe Model
- Push-Pull Model
- Exclusive Pair Model

Request-Response Model

- the client sends requests to the server and the server responds to the requests
- the server **receives a request, it decides how to respond, fetches the data, retrieves resource representation, prepares the response, and then sends the respons**e to the client.
- Request-response is a **stateless communication** model and each request-response pair is independent of others.
  - HTTP works as a request-response protocol between a client and server. A web browser may be the client, and an application on a computer that hosts a web site may be the server.

Request-Response Model

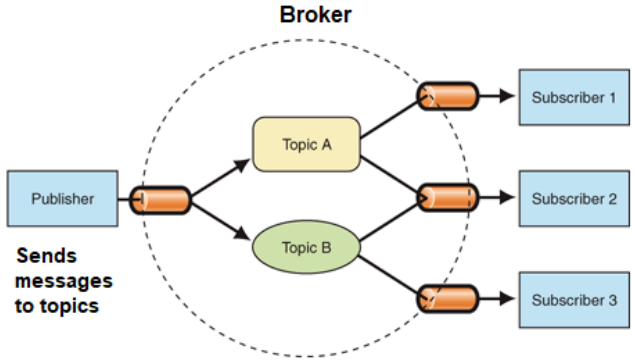


Request-Response Communication Model

Publish-Subscribe Model

- It has publishers, brokers and consumers
- Publishers are the source of data. Publishers send the 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 receive data for a topic from the publisher, it sends the data to all the subscribed consumers.

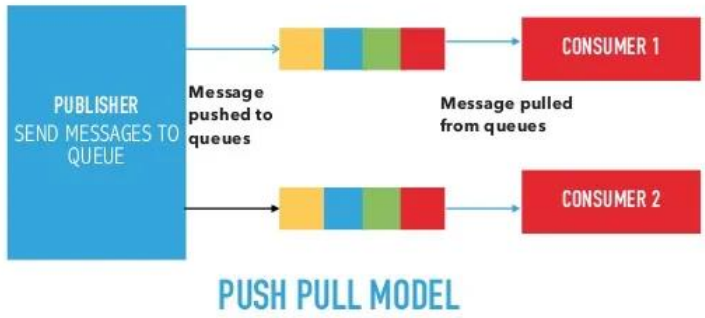
Publish-Subscribe Model



Push-Pull Model

- Push-Pull is a communication model in which the data producers push the data to queues.
- the consumers Pull the data from the Queues.
- Producers (communication producers) do not need to be aware of the consumers (communication consumers).
- Queues help in decoupling the messaging between the Producers and Consumers.
- Queues also act as a buffer which helps in situations when there is a mismatch between the rate at which the producers push data and the rate at which the consumer pull data.

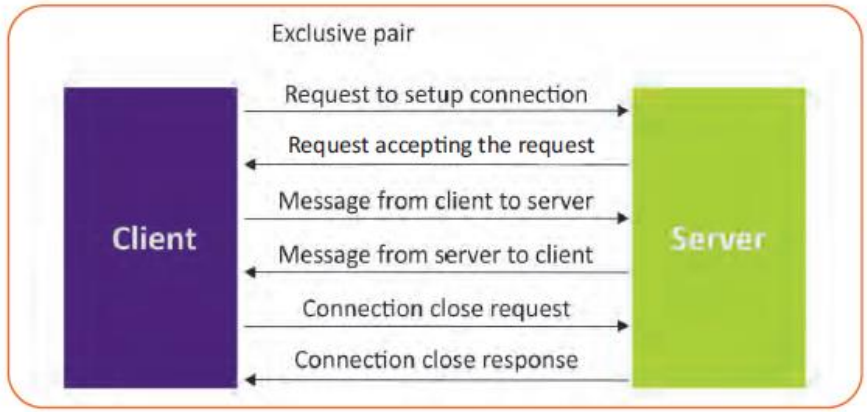
Push-Pull Model



Exclusive Pair Model

- Exclusive Pair is a bidirectional, fully duplex communication model that uses a persistent connection between the client and server.
- Connection is setup it remains open until the client sends a request to close the connection.
- Client and server can send messages to each other after connection setup.
- Exclusive pair is state-full communication model and the server is aware of all the open connections.

Exclusive-Pair



IoT Communication APIs

- Two APIs for IoT communication
  - REST-based Communication APIs
  - Web-Socket based Communication APIs

REST-based Communication API

- Representational State Transfer (REST) architecture.
- Design for web services web APIs towards system's resources processing like states of the resources are addresses and transferred.
- Follows **request and response** communication model.

characteristics

- The abstract information in REST is resource
  - A resource is
    - DOCUMENT
    - IMAGE
    - TEMPORAL SERVICE
    - COLLECTION OF RESOURCES
    - NON-VIRTUAL OBJECT
- Identifies the object or resource using resource identifier.
- Resource representation
  - State of the resource at any particular timestamp.
  - Data,meta data, hyperlinks, media type.
- Resource Methods : to perform desired transaction for the resource
  - GET/PUT/POST/DELETE

REST Guiding Principles or constraints

- Client-Server
- Stateless
- Cacheable
- Uniform interface
- Layered System
- Code on demand

Client-Server constraint

- The concerns are separated to both client and servers
- Clients should not bother about the storage of data
- Servers should not bother about the user interface
- Concerns are developed independently and should be updated.

Stateless constraint

- Information maintain to understand about the request at server.
  - The context of client cannot be stored at server.
- This state is maintained as sessions at client.

Cache-able constraint

- Storing the data of a response for a request is cache-able.
  - cache-able or non cache-able (storing data) either implicitly or explicitly.
- If cache-able
  - Client cache has the permission to store the equivalent data of requests
  - It can be used later.
  - It improves the efficiency and scalability
- Non cache-able
  - No permission to cache the data at client side.

Layered System

- Separating the components of system as layers.
- This constraint is implemented in such a way that, one component can not see beyond its immediate layer with which it is going to interact.
  - For example, client did not reveal to the server about its connectivity as it is connected directly or indirectly.
- This provides **scalability** by adding the components of the system at different layers or it can be add the functionalities at immediate server or at intermediate server that are in between the path.
  - So the intermediate server may respond for client requests instead of end server.

Uniform interface

- Common method of communication need to be followed between client and server.
- All the resources of requests need to be identified by common web based representation (Like URI's).
- Representation of resource can be created, updated, deleted and added with proper permission.
- Each message should provide enough information regarding this process. (eg: GET,PUT,DELETE and POST etc)

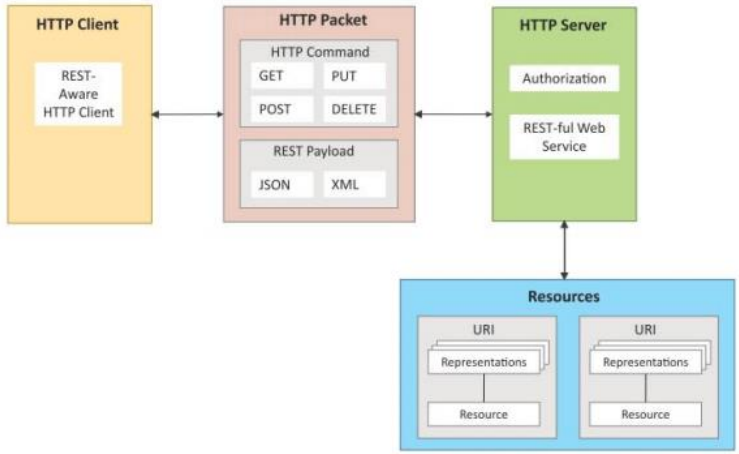
Code on demand constraint

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

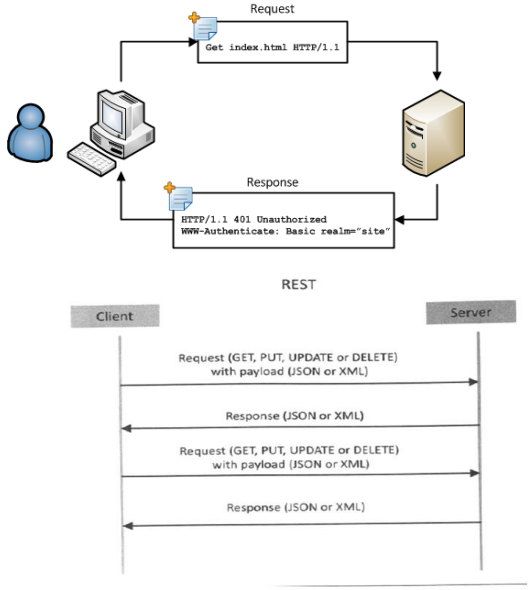
RESTful web service

- It is a " Web API " implemented using HTTP and REST principles.
- REST is most popular IoT Communication APIs

RESTful web service



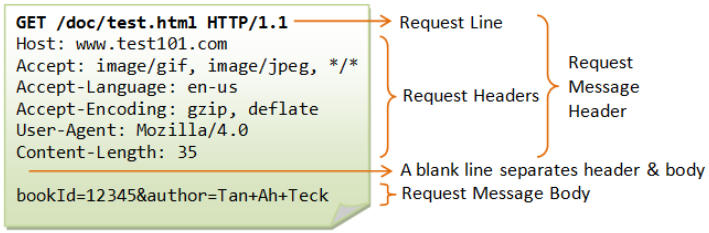
Request Response model by REST API



HTTP Request Methods and Actions

HTTP Method	Resource Type	Action	Example
GET	Collection URI	List all the resources in a collection	http://example.com/api/tasks/ (list all tasks)
GET	Element URI	Get information about a resource	http://example.com/api/tasks/1/ (get information on task-1)
POST	Collection URI	Create a new resource	http://example.com/api/tasks/ (create a new task from data provided in the request)
POST	Element URI	Generally not used	
PUT	Collection URI	Replace the entire collection with another collection	http://example.com/api/tasks/ (replace entire collection with data provided in the request)
PUT	Element URI	Update a resource	http://example.com/api/tasks/1/ (update task-1 with data provided in the request)
DELETE	Collection URI	Delete the entire collection	http://example.com/api/tasks/ (delete all tasks)
DELETE	Element URI	Delete a resource	http://example.com/api/tasks/1/ (delete task-1)

Request and Response format of web page

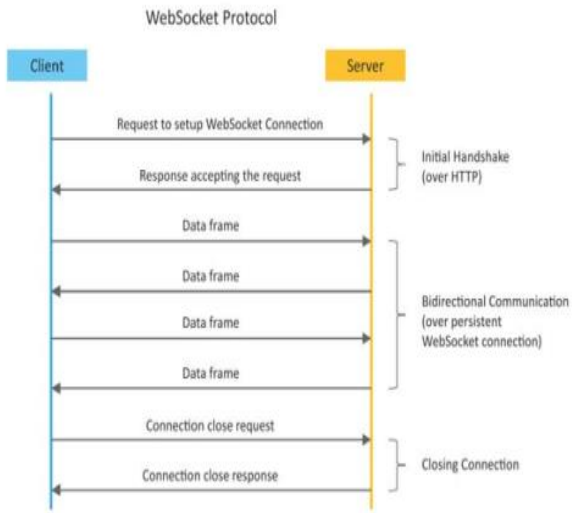


Difference between REST & HTTP

- The REST architectural style, data and functionality are considered resources and are accessed using Uniform Resource Identifiers (URIs).
- The resources are acted upon by using a set of simple, well-defined operations.
- The clients and servers exchange representations of resources by using a standardized interface and protocol – typically HTTP

Web Socket based Communication API

- Bi-directional, full duplex communication.
- It follows the exclusive pair communication model.
- New connection need not be setup for each request due its full duplex communication.
- It performs
  - Request response hand shake procedure
    - Web-socket hand shake
  - Data communication
  - Connection termination.
- Webscoket API reduce the network traffic and latency due to its reduced connection set up for each request.
- Suitable for low latency and high through put requirements.



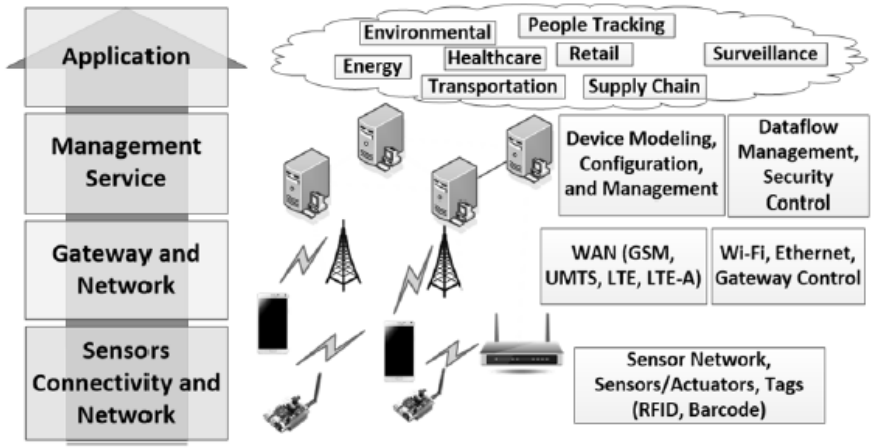
Physical Design of IoT

- Things are Node device which have unique identities and can perform **remote sensing, actuating** and **monitoring** capabilities.
- IoT Protocols helps Communication established between things and cloud based server over the Internet.

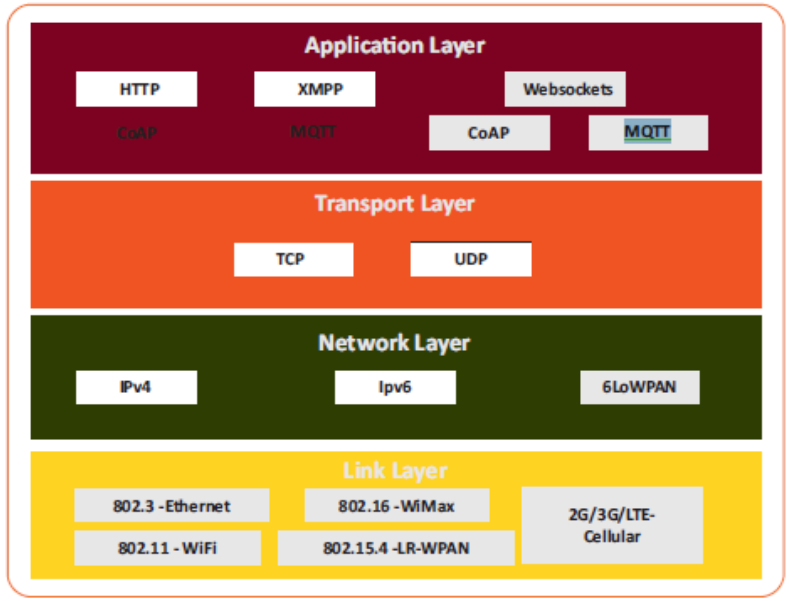
IoT Protocols

- IoT protocols help to establish Communication between IoT Device (Node Device) and Cloud based Server over the Internet.
- It help to sent commands to IoT Device and received data from an IoT device over the Internet.

IoT Architecture



Communication Protocol



Communication Protocols

Contd.,

- IoT connects various devices through Internet and devices talk to each other through a communication channel. The defined communication channel is called as 'Communication Protocol'.
- The IoT specific protocols cover the wireless protocols like ZigBee, RFID, Bluetooth and BAC net and the protocol standards such as 802.15.4e, 6LoWPAN, RPL, CoAP etc.

Link Layer

- how data is physically sent over the network's.
- It determines how the packets are coded and signaled by the hardware device over the medium to which the host is attached (eg. coxial cable).

Link layer protocols

- **802.3 – Ethernet**
  - Ethernet is a set of technologies and protocols that are used primarily in LANs
  - standardized in 1980s by IEEE 802.3 standard**.**
  - sub-layer of the data link layer for wired Ethernet networks
  - IEEE 802.3 defines the physical layer and the medium access control (MAC).
  - This is generally a local area network (LAN) technology with some wide area network (WAN) applications.
  - Physical connections are made between nodes and/or infrastructure devices (hubs, switches, routers) by various types of copper or fiber cable.
  - Ethernet is classified into two categories: classic Ethernet and switched Ethernet.

Link layer protocols

- **802.16 – Wi-Max**
  - is a standard for Wireless Metropolitan Area Networks (WMANs) that has been developed by working group **of IEEE 802**
  - point-to-multipoint broadband wireless access

Link layer protocols

- **802.15.4 -LR-WPAN**
  - A collection of standards for Low-rate wireless personal area network.
  - The IEEE's 802.15.4 standard defines the MAC and PHY layer used by, networking specifications such as Zigbee®, 6LoWPAN, Thread, WiSUN and MiWi™ protocols.
  - The standards provide low-cost and low-speed communication for power constrained devices.

Link layer protocols

- **2G/3G/4G- Mobile Communication (Cellular communications)**
- These are different types of telecommunication generations.
- IoT devices are based on these standards can communicate over the celluer networks.

Network Layer

- Responsible for sending of IP datagrams from the source network to the destination network.
- Network layer performs the host addressing and packet routing.
- We used IPv4 and IPv6 for Host identification.
- IPv4 and IPv6 are hierarchical IP addressing schemes.

IPv4

- An **Internet Protocol address** (**IP address**) is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication.
- An IP address serves two main functions: host or network interface identification and location addressing.
- Internet Protocol version 4 (IPv4) defines an IP address as a 32-bit number

Drawback of IPv4

- Because of the growth of the Internet and the depletion of available IPv4 addresses, a new version of IP (IPv6), using 128 bits for the IP address,

IPv6 : Internet Protocol version 6 (IPv6)

- recent version of the Internet Protocol (IP), the communications protocol
- provides an identification and location system for computers on networks and routes traffic across the Internet.
- Pv6 was developed by the Internet Engineering Task Force (IETF) to deal with the long-anticipated problem of IPv4 address exhaustion.
- IPv6 is intended to replace IPv4.
- IPv6 uses a 128-bit address, theoretically allowing 2^{128}, or approximately 3.4×10^{38} addresses

6LoWPAN

- *IPv6 over Low-Power Wireless Personal Area Networks (*6LoWPAN).
- combines the latest version of the Internet Protocol (IPv6) and Low-power Wireless Personal Area Networks.
- allows for the smallest devices with limited processing ability to transmit information wirelessly using an internet protocol.
- 6LoWPAN can communicate with 802.15.4 devices as well as other types of devices on an IP network link like WiFi.

Transport Layer

- provides functions such as
  - error control,
  - segmentation,
  - flow control and
  - congestion control.
- provide end-to-end message transfer capability independent of the underlying network.

TCP (Transmission Control Protocol)

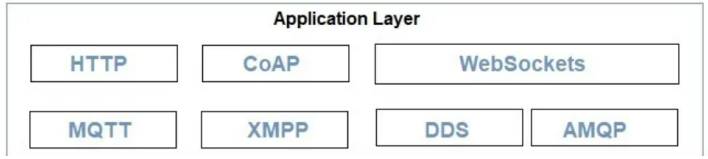
- defines how to establish and maintain a network conversation through which application programs can exchange data.
- defines how computers send packets of data to each other
- The Internet Engineering Task Force (IETF) defines TCP.
- Together, TCP and IP are the basic rules defining the Internet

User Datagram Protocol (UDP)

- UDP is a part of Internet Protocol suite, referred as UDP/IP suite.
- it is unreliable and connectionless protocol.
- No need to establish connection prior to data transfer.

Application Layer

- Define how the applications interface with the lower layer protocols to send over their network.



HTTP : *Hypertext Transfer Protocol (HTTP)*

- protocol for transmitting hypermedia documents, such as HTML.
- designed for communication between web browsers and web servers
- follows a classical client-server model with uni-directional, with a client opening a connection to make a request, then waiting until it receives a response.
- HTTP is a stateless protocol, meaning that the server does not keep any data (state) between two requests.
- Often based on a TCP/IP layer

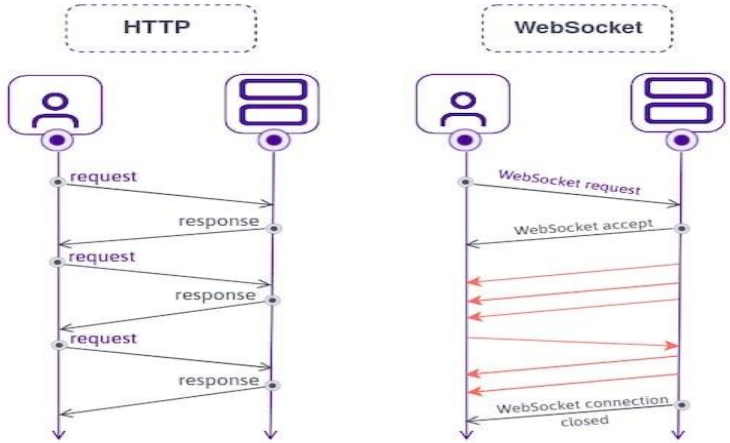
drawbacks

- The HTTP is made for two systems communicating to each other at a time, not more, so it is time and energy-consuming to connect several sensors to get information.
- The HTTP is unidirectional, made for one system (client) to be sending one message to another one (server). This makes it quite hard to escalate an IoT solution.
- Power consumption: HTTP relies on Transmission Control Protocol (TCP), which requires a lot of computing resources, so it is not suitable for battery-powered applications.

Constrained Application Protocol(CoAP)

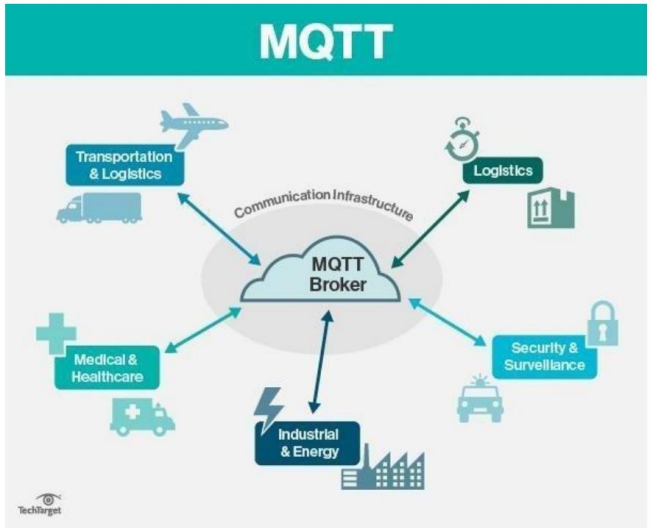
- is a specialized Internet Application Protocol for constrained devices.
- It is a simplification of the HTTP protocol running on UDP, that helps save bandwidth,but no guarantee on delivery.
- It enables low power devices to communicate over the Internet.
- Follows client/server architecture which follows request and response model.
- The protocol is especially targeted for constrained hardware such as 8-bits microcontrollers, low power sensors and similar devices that can't run on HTTP or TLS.
- designed for use between
  - devices on the some constrained network (e.g., low-power, lossy networks, embedded devices with limited power, memory, and processing resources).
  - Used for limited networks with low bandwidth and low availability.

Differences between HTTP & WebSocket

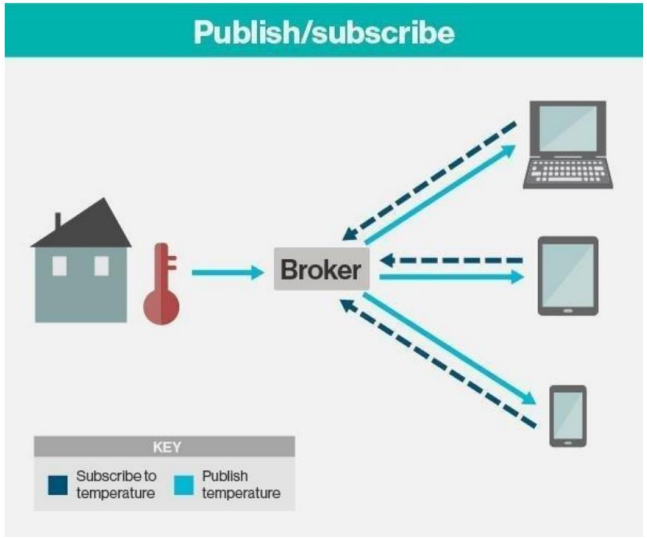


Message Queuing Telemetry Transport (MQTT)

- MQTT is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol.
- designed as an extremely lightweight publish/subscribe messaging transport and useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium.
- MQTT protocol runs on top of the TCP/IP networking stack.
- When clients connect and publish/subscribe, MQTT has different message types that help with the handshaking of that process.
- The MQTT header is two bytes and **first byte is constant**. In the first byte, you specify the type of message being sent as well as the **QoS level, retain, and DUP (duplication) flags**. The second byte is the remaining length field



Simple Use case



Smart Factory

XMPP : Extensible Messaging and Presence Protocol (XMPP)

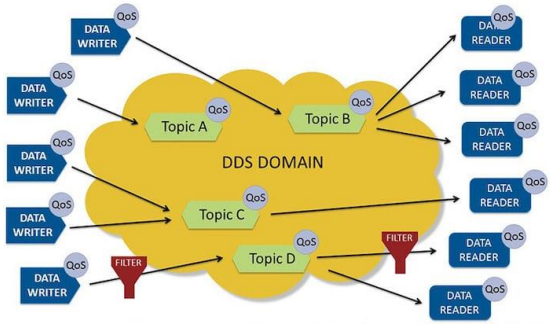
- is a communication protocol for message-oriented middleware based on XML
- It enables the near-real-time exchange of structured yet extensible data between any two or more network entities.
- Originally named **Jabber**
- Developed by the eponymous open-source community in 1999 for near real-time instant messaging (IM), presence information, and contact list maintenance.
- It is an open-source, decentralized, secure protocol to exchange XML messages.
- has been used for publish-subscribe systems.
- Applications:
  - VoIP, video, file transfer, gaming, the Internet of Things (IoT) applications such as the smart grid, and social networking services

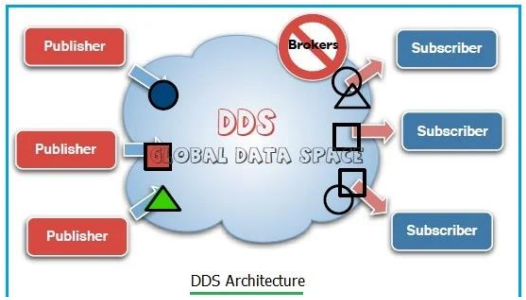
DDS: Data Distribution Service

- a middleware protocol and API standard for data-centric connectivity from the Object Management Group (OMG).
  - middleware is the software layer that lies between the operating system and applications
- integrates the components of a system together, providing low-latency data connectivity, extreme reliability, and a **scalable architecture that business and mission-critical Internet of Things (IoT) applications need**.
- It enables the various components of a system to more easily communicate and share data.
- It enables data exchange via publish-subscribe methodology without broker.
- It simplifies the development of distributed systems by **letting software developers focus on the specific purpose of their applications rather than the mechanics** of passing information between applications and systems.
- It uses multicasting to bring high quality QoS to the applications.
- DDS protocol can be deployed from low footprint devices to cloud.

Operation

- DDS follows a publish-subscribe methodology, with the main difference being that there are no brokers.
- all publishers (i.e., temperature sensors) and subscribers (i.e., mobile phones) are all connected to the same network. This network is known as Global Data Space (GDS).
- it interconnects each node with all the other ones to avoid bottlenecks





DDS Architecture

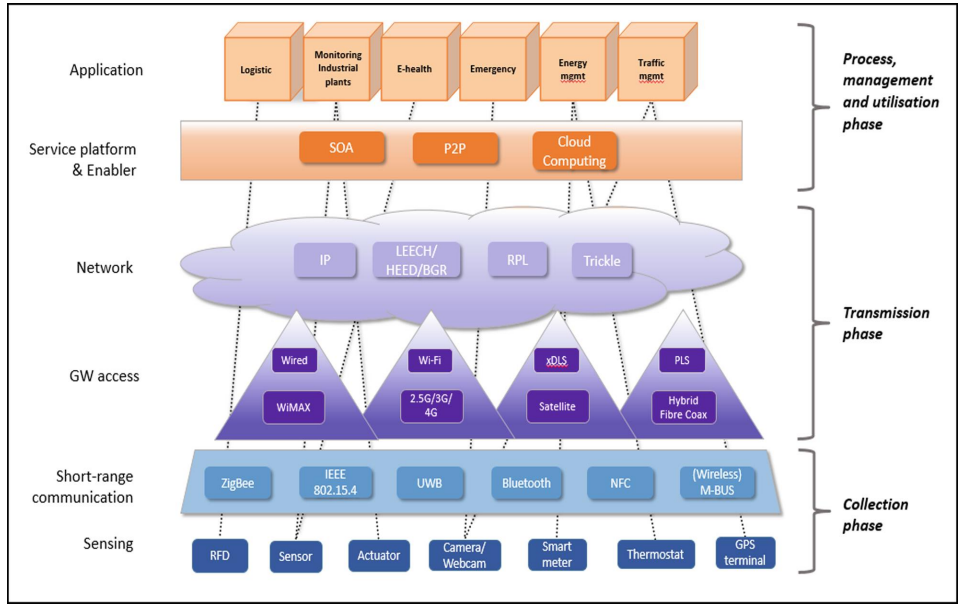
Advanced Message Queuing Protocol (AMQP)

- IoT protocols consist of a hard and fast of components that route and save messages within a broker carrier.
- AMQP protocol enables patron programs to talk to the dealer and engage with the AMQP model.
  - The Patron Program was introduced in 2017 as a way for businesses and individuals to financially support the performing arts performances. Patrons are recognized in publications, and free tickets based on the level of support.
- It is TCP Based.
- AMQP has the following three additives
  - **Exchange:** Receives messages from publisher primarily based programs and routes them to 'message queues'.
  - **Message Queue:** Stores messages until they may thoroughly process via the eating client software.
  - **Binding:** States the connection between the message queue and the change.
- **AMQP allows both the client/broker(publisher/subscriber) and client/server architectures.**.

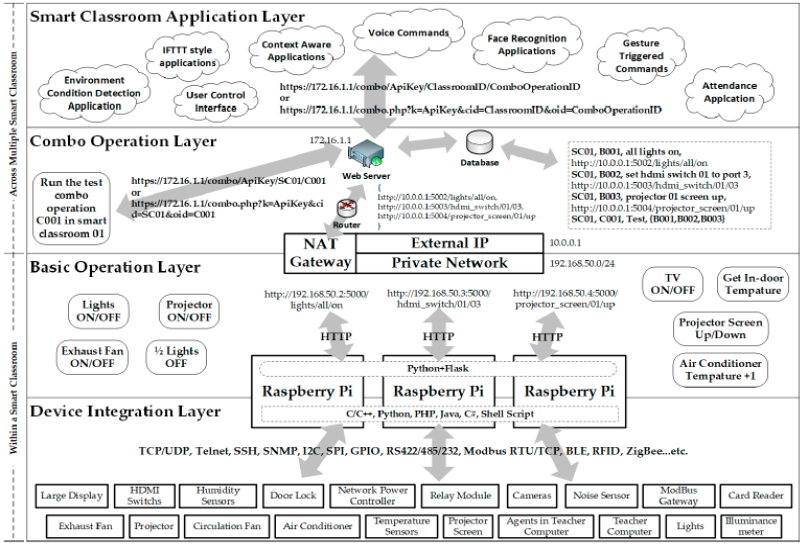
OPC Unified Architecture (OPC UA)

- It is a standard made for industrial communication, strongly oriented to guarantee interoperability between manufacturers, operating systems, and programming languages.
- The OPC foundation reported that many industrial vendors are currently (as of 2022) adopting the OPC UA as the open standard.
- the OPC UA is a transport-agnostic protocol, so it supports both of the previously used architectures: request/response (such as WebSocket or HTTP), as well as the publish/subscribe (such as the MQTT).

IoT Model



IoT Architecture for Smart classroom Integration model



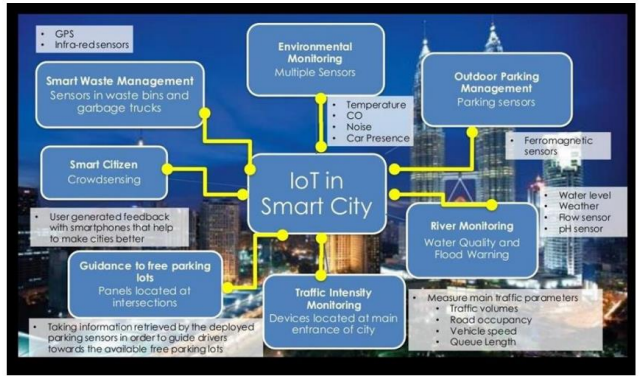
Examples

- Case study on smart lighting
- Case study on smart home
- Case study on smart cities
- Case study on smart factory
- Case study on smart energy/smart grid

Smart Home

- Identifying and indicating abnormalities from the home and kitchen
- Understanding when to turn on lights, ovens, heater on the way to home from work.
- Switch on the favourite music station while you are at your doorsteps
- Recommending the best time in your day to start the washing machine ad cleaning dishes for kitchen.
- Switch off lights when you vacate the room
- Smart lock
  - Lock the door when you vacate the room
- Smart refrigerator
  - Notifying about grocery items in the freezer.

IoT in Smart City



Smart city

- fitness machines at your local gym programming notify your personal workout when you leave the locker room
- Smart Surveillance
- Smart lighting
- Smart maintenance of public resources
- Smart waste management.
- multi-modal Integrated Ticketing and Automated Fare Collection

Automatic vehicle counting

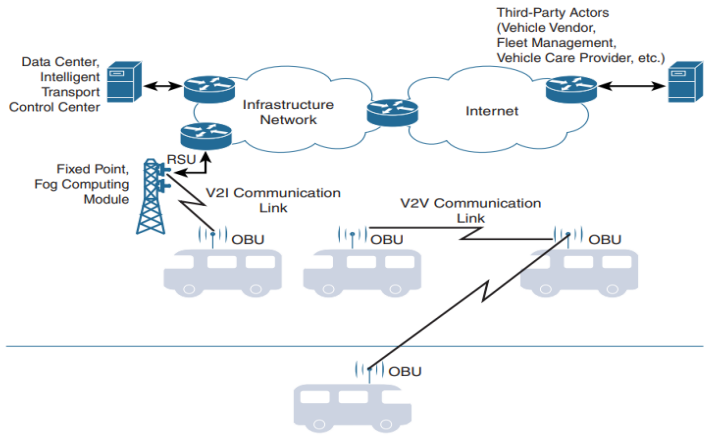

Automatic vehicle counting

- The video sequences provide glimpse of the vehicular traffic in two urban Indian cities
- vehicle detection and counting is challenging task.
  - presence of large number of different type of vehicles along with absence of systemized lane rule procedure

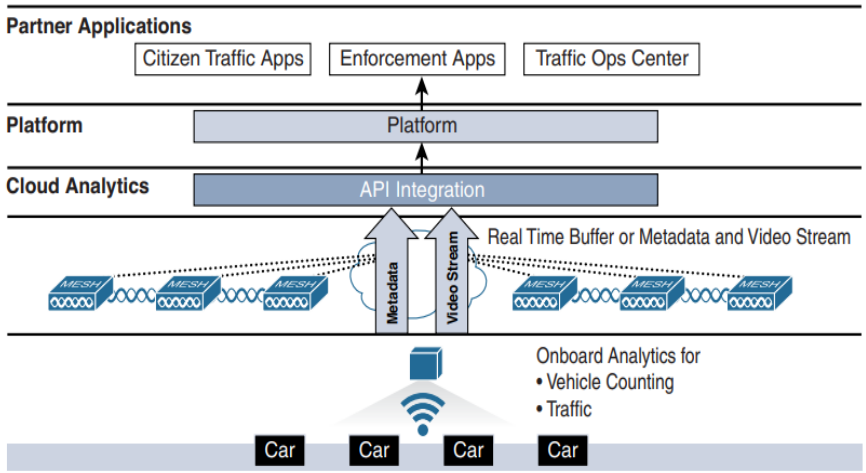
Automatic vehicle counting

- The process starts from extracting a single image frame from the real time CCTV footage.
- the foreground image obtained after applying Gaussian background subtraction through Open CV in figure

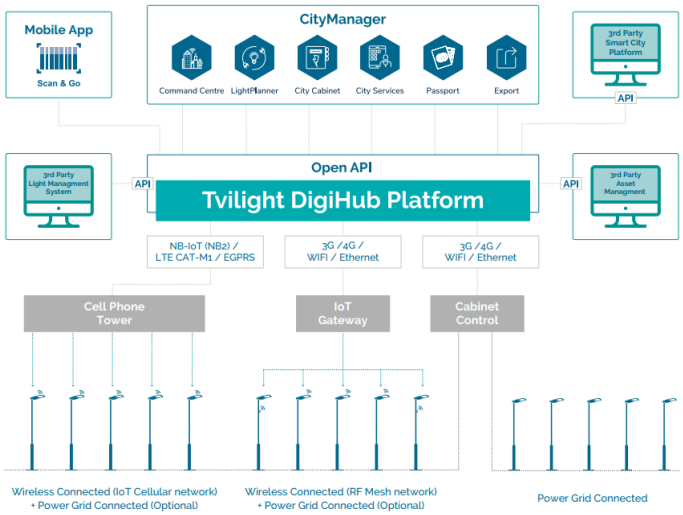

Vehicle to vehicle communication using DSRC



Smart Traffic control Architecture



Smart Light



Smart integration

- Interoperable devices to integrate
- Interoperable systems connecting together
- Interoperable communication platform

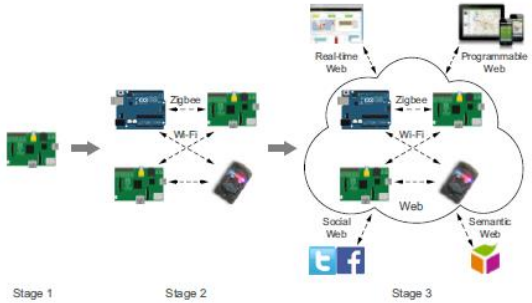
Smart industry

- Notifying about machinery wear and tear information
- Notifying the abnormalities of machinery parts.
- Notifying the time period of maintenance period to machinery.

Connectivity Stages

the three types of connectivity (stages)

1. lonely device that can sense and interact with its surroundings but doesn't have any connectivity
2. the device supports at least one communication protocol and can talk to other devices to form a small network of devices
3. devices can be connected to the wider web ecosystem so that any application or service can talk to these devices over the internet



Connecting Things

Networking topologies or Physical structure of the IoT

- Point to point
- Star Networks
- Mesh Networks

Point-to-point

when any two devices establish a direct connection and start talking to each other is called point-to-point

used in the context of wearable devices:

- you synchronize **your fitness tracker with your mobile phone** over Bluetooth by pairing the two devices
- a thermostat can create a point-to-point network called Wi-Fi ad hoc mode, where you can connect with your mobile phone and send to your thermostat the credentials and configuration of your home network.

STAR Networks

several nodes communicate with a single central node and might not be aware of other nodes in the network.

This model is also often used as a star of stars, where each central node is in turn connected to another nearby central node.

What is drawback here?

Mesh Networks

the most complex: mesh networks.

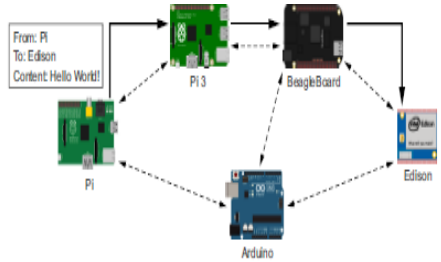
In a mesh network, there are no central nodes because any node . in the network is able to forward messages from one node to another

the Pi can use the nearby devices as intermediate hops, called relays, to forward the message to its destination.

the network more robust against individual node failures. For instance, if the Arduino fails, the Pi could still communicate with the Edison by going through all the other devices

Not a point-to-point connection,but all are connected through the internet by hopping the messages from one router to another until it reaches the destination.

Example : monitoring the pollution level in a remote forest



Smart or Intelligent transportation system

- Improving the mobility methods while dropping the congestion on streets.
- Emergency response system on congestion of streets with enhanced efficiency and safety.
- Monitoring the transportation system in real time with minimal delay.
- Management of car navigation system
- Traffic signal control system
- Automatic number plate recognition system with speed camera monitoring application
- Smart parking

References

- Arshdeep Bahga, Vijay Madisetti, “Internet of Things: A hands-on Approach”, University Press, 2015.
- IT-Ites SSC NASSCOM Participant handbook, “Foundation skills in Internet of Things(IoT)”
- **IoT (Internet of Things) Wireless & Cloud Computing Emerging Technologies by Jong-Moon Chung,Yonsei University.**