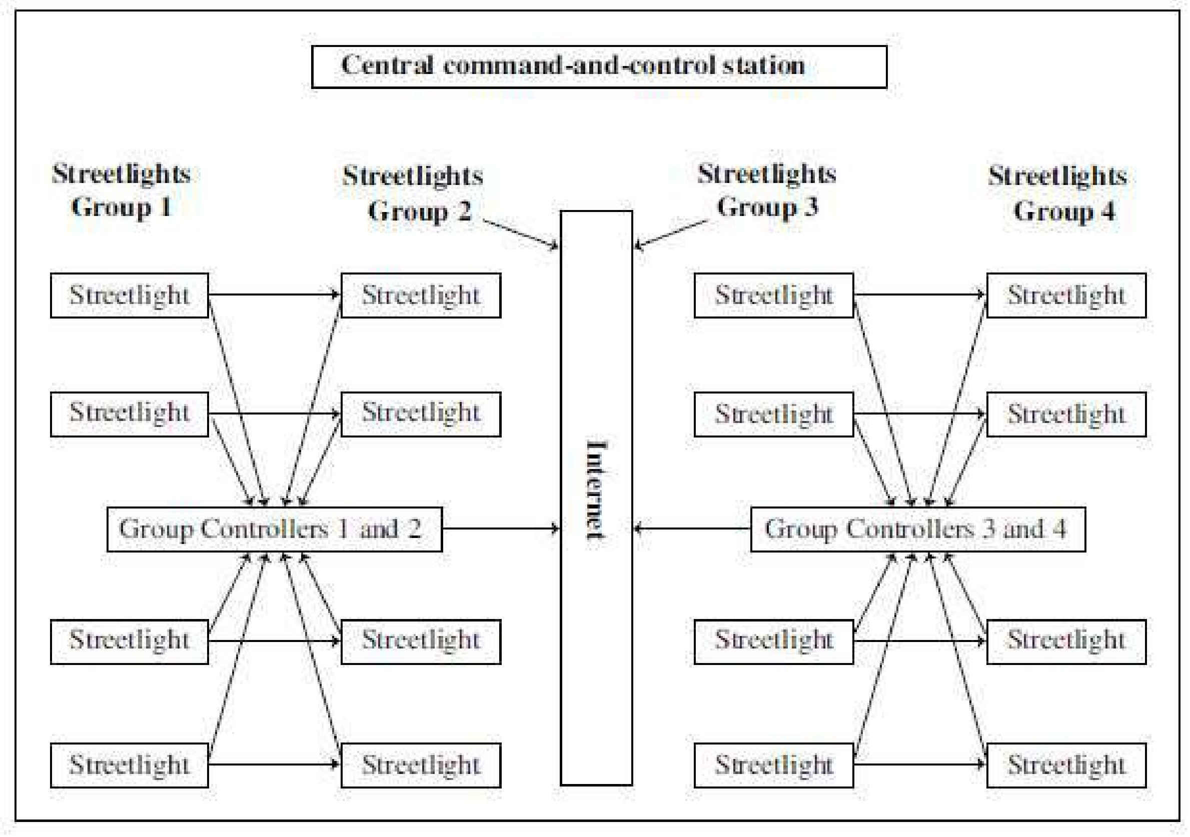
Module 1

Introduction : Definition , Internet of Things IoT Architectural view, IoT Technology, M2M Communication, Success Factors of Internet of Things, IoT Application Areas , IoT Functional View, Design Principles for connected Devices, Communication Technologies

Internet Of Things

- It’s a concept which enables communication between internetworking devices and applications, whereby physical objects or ‘things’ communicate through internet.
- A network of physical objects or ‘things’, sending, receiving or communicating information using internet/other communication technologies and network, just as computers, mobiles, tablets do – and thus enabling the monitoring, coordination, controlling process across internet or another data network

Streetlights in a city can be made to function like living entities through sensing and computing using tiny embedded devices that communicate and interact with a central control-and-command station through the Internet.

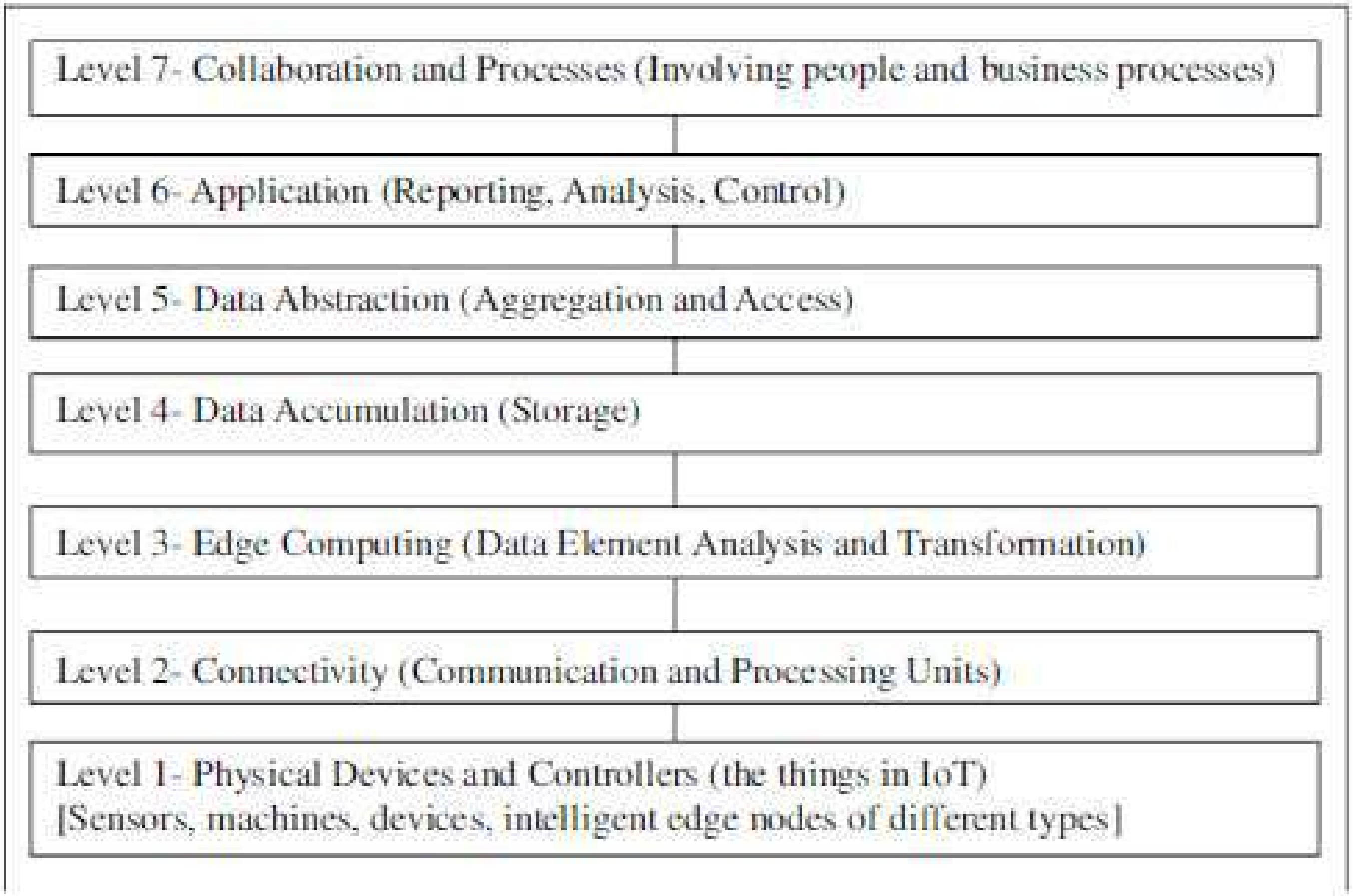


IoT Architectural View

An architecture has the following features:

- The architecture serves as a reference in applications of IoT in services and business processes.
- A set of sensors which are smart, capture the data, perform necessary data element analysis and transformation as per device application framework and connect directly to a communication manager.
- A set of sensor circuits is connected to a gateway possessing separate data capturing, gathering, computing and communication capabilities. The gateway receives the data in one form at one end and sends it in another form to the other end.
- The communication-management subsystem consists of protocol handlers, message routers and message cache.
- This management subsystem has functionalities for device identity database, device identity management and access management.
- Data routes from the gateway through the Internet and data centre to the application server or enterprise server which acquires that data.
- Organisation and analysis subsystems enable the services, business processes, enterprise integration and complex processes

Architectures are based on reference models. A typical reference model developed by CISCO is given below:



Level 7- Collaboration and Processes (Involving people and business processes)

Level 6- Application (Reporting, Analysis, Control)

Level 5- Data Abstraction (Aggregation and Access)

Level 4- Data Accumulation (Storage)

Level 3- Edge Computing (Data Element Analysis and Transformation)

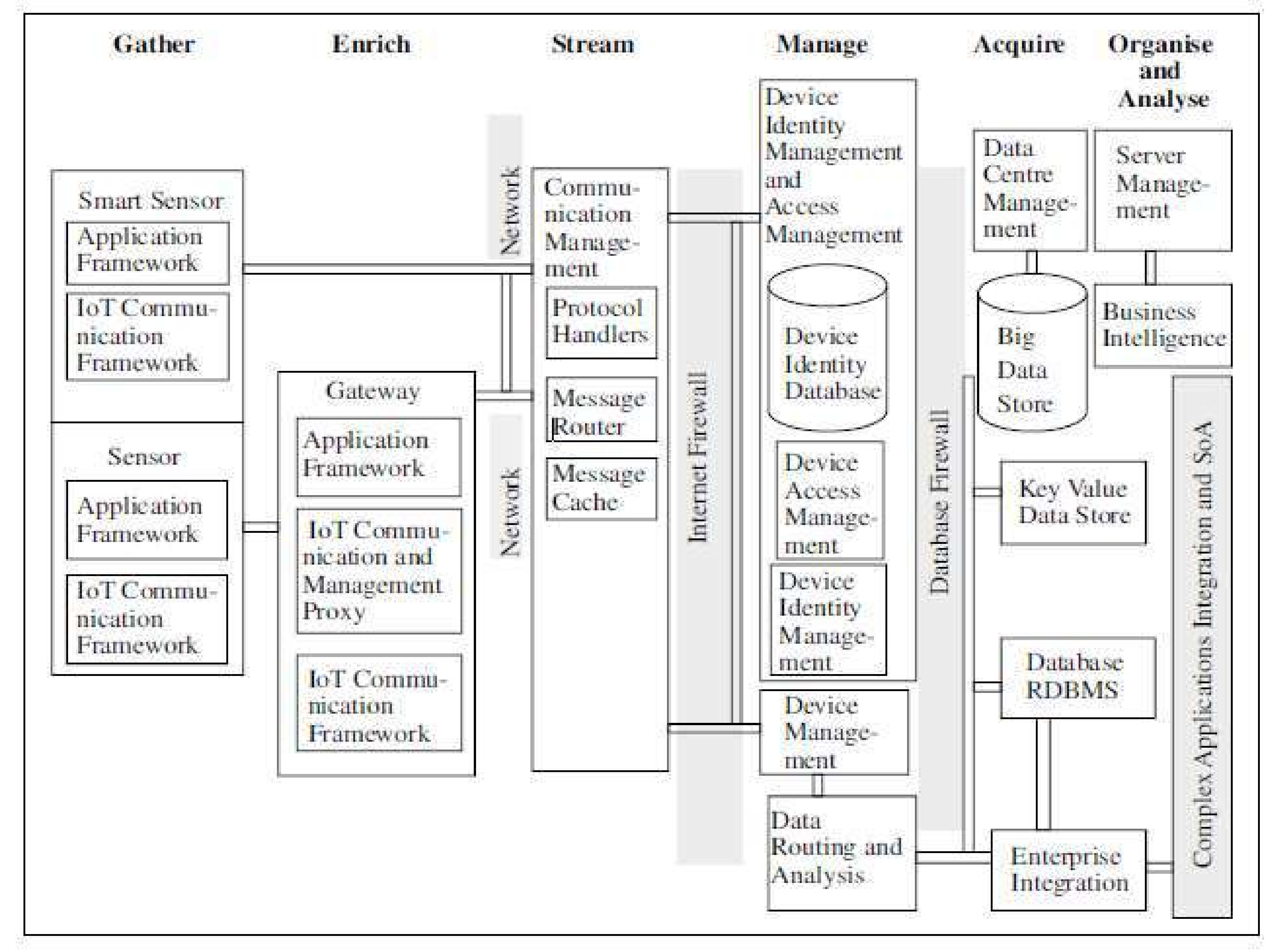
Level 2- Connectivity (Communication and Processing Units)

Level 1- Physical Devices and Controllers (the things in IoT)
[Sensors, machines, devices, intelligent edge nodes of different types]

CISCO seven leveled reference model

ARCHITECTURAL VIEW developed by Oracle based on the conceptual framework:

Gather + Enrich + Stream + Manage + Acquire + Organise and Analyse
= **Internet of Things with connectivity to data centre, enterprise or cloud server**



- P2413 – A new standard for IoT Architecture is being developed by IEEE

Technology Behind IOT

- Hardware – Arduino, Raspberry Pie, Intel Galileo, Intel Edison, ARM mBed…
- Integrated Development Environment (IDE) for developing device software, firmware, API (Application programming interface)
- Protocols – RPL, CoAP, RESTful HTTP, MQTT…
- Communication – Powerline Ethernet, RFID, NFC, Zigbee, Bluetooth, WiFi, WiMax, 2G/3G/4G…
- Network Backbone – IPv4, IPv6, UDP…
- Software – RIOT OS, Contiki OS, Thingsquare, Eclipse IOT …
- Internet cloud platforms – Sense, Nimbits, AWS IoT, TCS Cup, IBM BlueMix, CISCO IoT … They are costly, but very flexible for users. Dedicated servers are cost effective.
- Machine learning algorithm and software. Eg: GROK from Numenta Inc. using machine intelligence to analyse the streaming data from clouds and uncover anomalies.

Server end technology

- Servers are critical components in IoT – eg: Application servers, enterprise servers, cloud servers, data centres…

Major Components of IoT System

1) Physical Object with embedded software into a hardware

- **Sensors – Smart sensors – Sensor actuator pairs**
  - Temp, Pressure sensors, accelerometers, gyroscopes, GPS sensors, proximity sensors, Magnetic field sensors etc…
  - Can give analog o/p – temp, pressure sensors
  - Can give digital o/p – touch sensors, proximity sensors, metal and water detectors
- **Control Units**
  - Microcontrollers like Atmega, ARM Cortex …
  - Has processor, memory, hardware interfaces, firmwares, timers. Communication interfaces…
  - May contain application specific functional circuits like ADC, DAC, PWM etc…

2) Communication modules

- Softwares – device API's & device interface for communication (CoAP, LWM2M, IPv4, IPv6…)
- Consists of protocol handlers, message queue and cache. Device message queue handles data in first in-first out manner

Major Components of IoT System

3) IOT Software

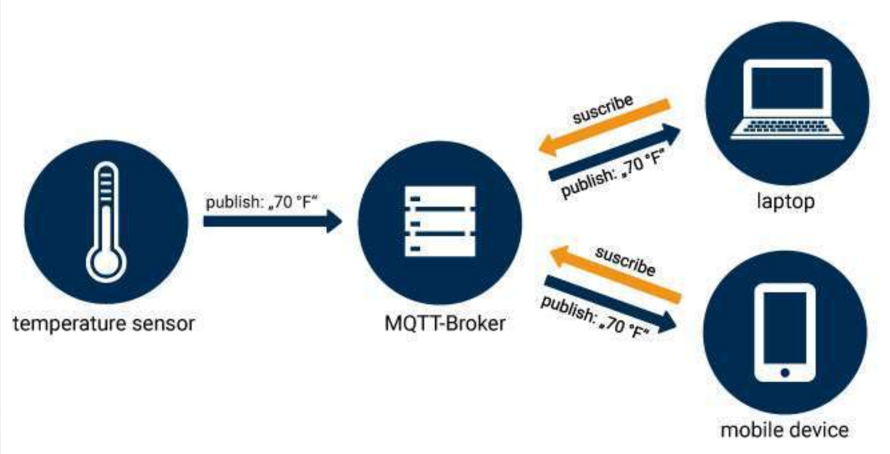
- **Middleware** (bridge b/w OS and end application on a network)
  - OpenIoT is an open sorce middleware – enables communication b/w sensor n/w and cloud base.
  - IoTSyS is anotherexample enabling smart communication b/w devices using protocols like IPv6, CoAP etc…
- **Operating System (** software for user interface)
  - *Raspbian* is a popular Raspberry Pi OS
  - RIOT is another example – supports ARM processors, Cortex, x86 PC's and TI MSP
  - AllJoyn is an open source OS by Qualcomm – available for Android, iOS, Windows, Linux
  - Spark is a distributed cloud based IoT OS
- **Firmware (**permanent software programmed into a read-only hardware memory)
  - Eg: Thingsquare Mist is an open source firmware

Development Tools & Open Source Frameworks for implementing IoT S/M

- Eclipse IoT – provides an open source implementation of various standards (MQTT CoAP, OMA-DM and OMA LWM2M) – Eclipse developed IoT programming language Lua
  - (Message Queuing Telemetry Transport, Constrained Application Protocol)
- Arduino development tools provide a set of software (including IDE) and Arduino programming language.
- Kinoma Software platform. Kinoma connect is a free app for Android and iOS.

MQTT

With **Message Queuing Telemetry Transport,** data is sent from a large number of machines to a single destination – the cloud – where the data can be analyzed, interpreted and forwarded. The cloud hosts an **MQTT broker** – an intermediary between machines and other machines and/or people. And this is an important distinction, as the machines do not communicate directly with each other, but through the broker.

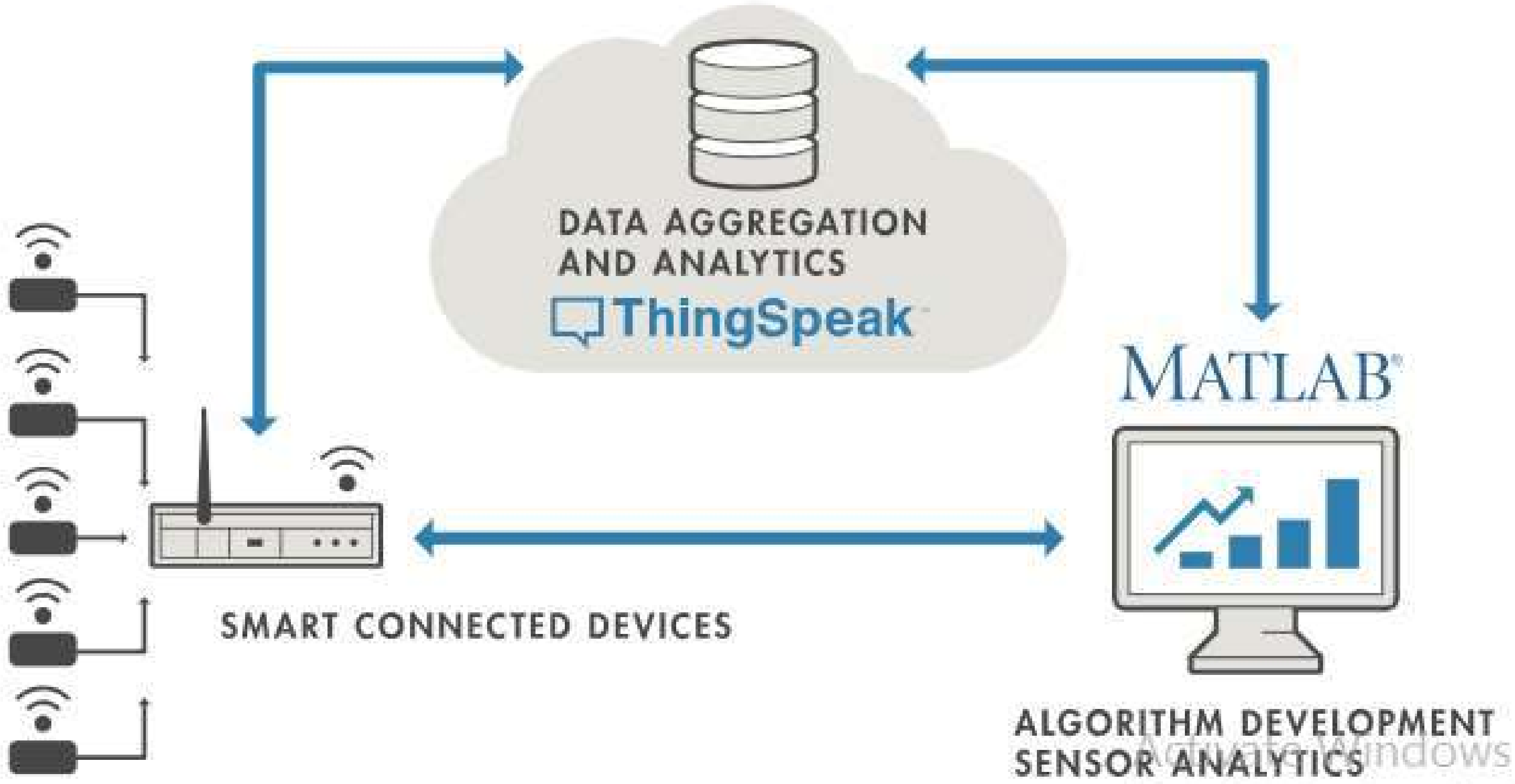


Mosquitto is an MQTT broker and part of the Eclipse Foundation and is a project of iot.eclipse.org

Application program Interface (API) and Device Interfacing components

Platforms and Integration Tools

- ThingSpeak – open data platform with an open API that can collect real time data, geological data, process it and visualize. Supports Arduino, Raspberry Pi, MATLAB data analytics…
- Nimbits – Cloud Platform
- IoT Toolkit
- SiteWhere


DATA AGGREGATION
AND ANALYTICS
ThingSpeak
MATLAB
SMART CONNECTED DEVICES
ALGORITHM DEVELOPMENT
SENSOR ANALYTICS

Sources of IoT

Popular Development Boards

- All development boards need an IDE (Integrated development Environment) for developing device software, firmware and API (Application programming interface)

Arduino Yun

- uses ATmega32u4 and includes WiFi, Ethernet, USB, Micro SD

Microduino

- small board compatible with Arduino – can be stacked with other boards.

Intel Galileo

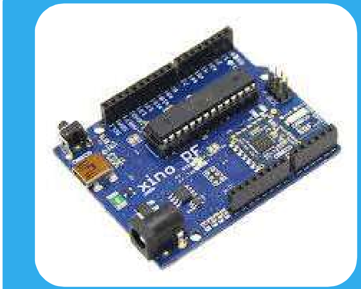
- Arduino certified development boards based on Intel x86 architecture – features Intel SOC X1000

Intel Edison

- It's a compute module – enables creation of smart IoT wearable and computing devices
- Has device internetworking and device to cloud communication

Beagle Board

- Has very low power consumption
- Has a card like computer and can run Android and Linux



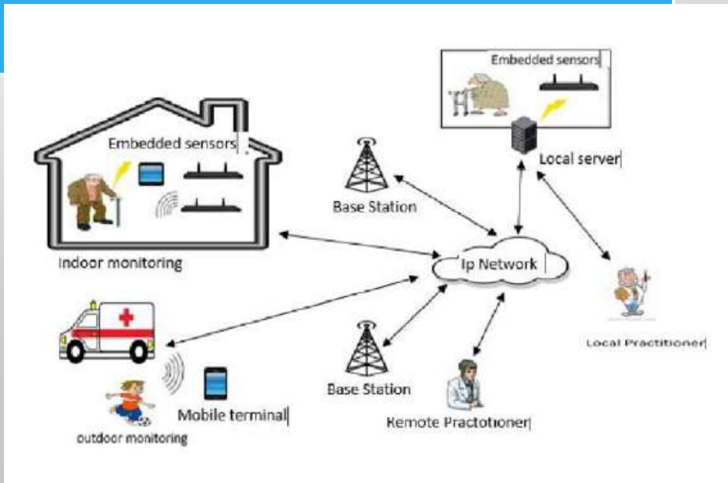
Raspberry Pi Wireless Inventors Kit (RasWIK)

- Enables Raspberry Pi WiFi connected devices

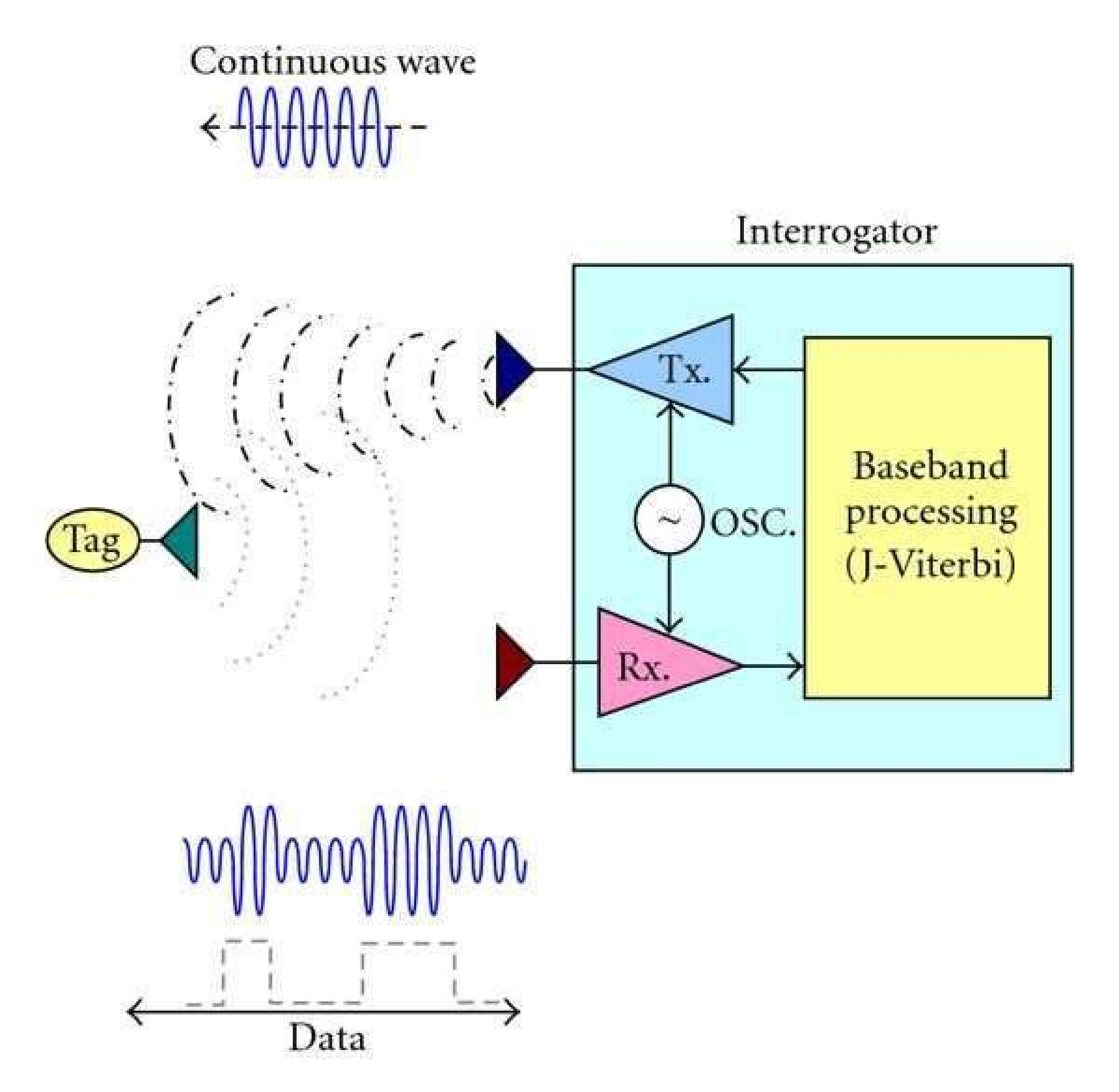
RFID – Radio-frequency identification - technology whereby digital data encoded in **RFID** tags or smart are captured by a reader via radio waves.

- Enables tracking and inventory control
- Identification in supply chains
- Access to buildings
- Road toll management
- Secured place entry
- RFID based temperature sensors
- New applications – factory design, anti counterfeiting in payments, quality management…

Wireless Sensor Networks (WSN) - A n/w in which, each sensor node connects wirelessly and has capabilities of computation for data compaction, aggregation and analysis, communication & networking.



- Sensors can be networked by wireless technology
- Analog/digial
- Can acquire data from remote locations
- Uses RF transceiver
- Temp, pressure, metal proximity etc…


Continuous wave
Interrogator
Tx.
Baseband
processing
(J-Viterbi)
OSC.
~
Tag
Rx.
Data

Machine to Machine - M2M COMMUNICATION

Process of communication of a physical object/device at machine with others of the same type, mostly for monitoring and control purposes

Each machine in M2M communication will have a smart device to enable this

Communication may be wired / wireless and use protocols like 6LowPAN, LWM2M, MQTT

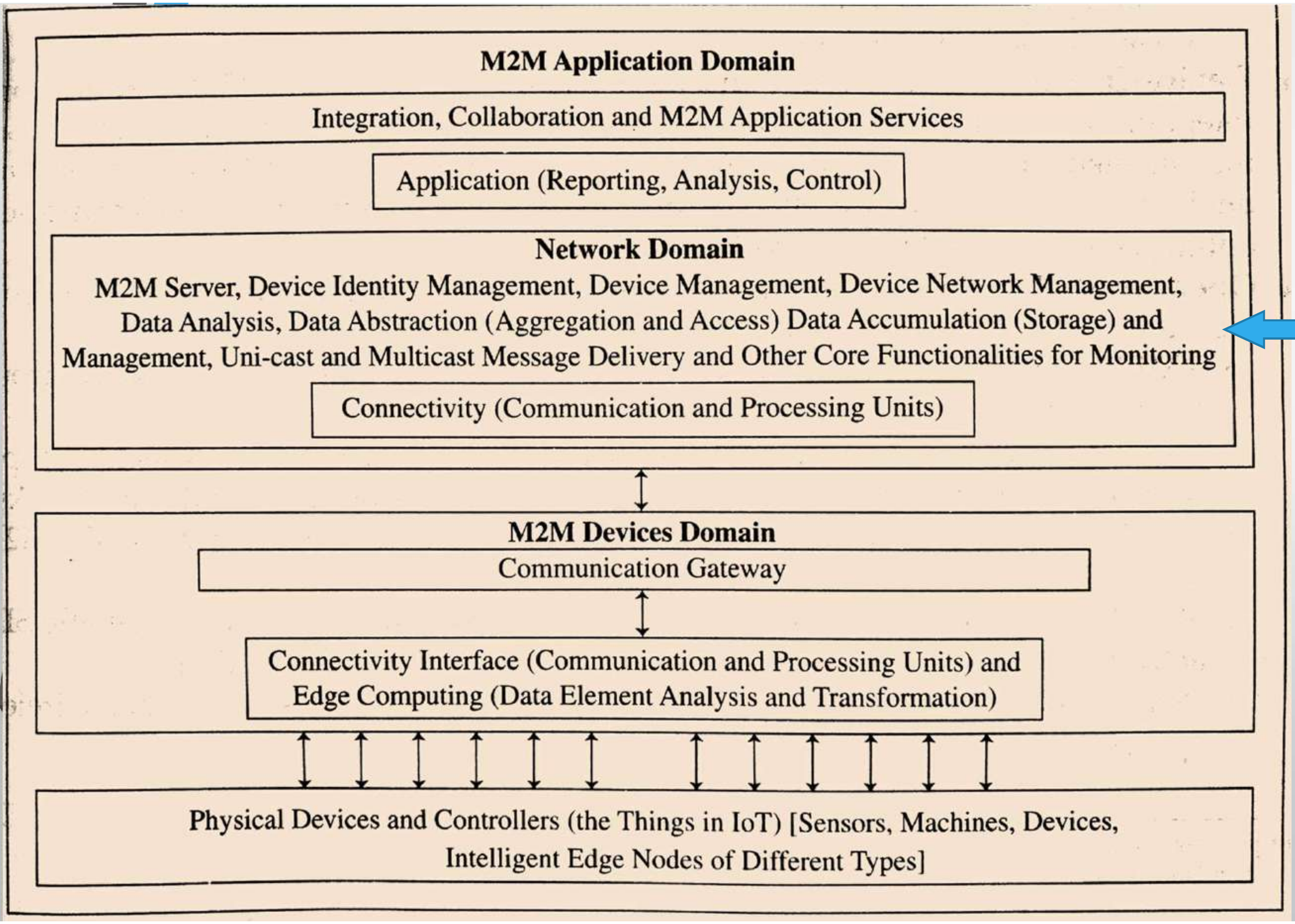
Each device will be assigned 48bits Ipv6 address

M2M vs IoT

- IoT integrates complex M2M communication with the cloud/network, analyses it and takes necessary actions
- M2M must deploy device to device and carry out coordination, monitoring, controlling of devices **without using Internet**.
- IoT **will use Internet**, servers, protocols and cloud based applications
- Example of M2M – coordinated movements of tools, robots, drones, refinery operations, sequential control in a production line etc…
- Applications – Industrial automation, logistics, smart grids, health and defence, **IIoT**
- IIoT – Industrial IoT – Manufacturing at multiple locations, railways, mining, agriculture, oil and gas, utilities, transportation etc… along with usage of internet, and softwares for analytics

M2M Architecture

- M2M Device Domain
- M2M Network Domain
- M2M Application Domain



M2M Application Domain

Integration, Collaboration and M2M Application Services

Application (Reporting, Analysis, Control)

Network Domain

M2M Server, Device Identity Management, Device Management, Device Network Management, Data Analysis, Data Abstraction (Aggregation and Access) Data Accumulation (Storage) and Management, Uni-cast and Multicast Message Delivery and Other Core Functionalities for Monitoring

Connectivity (Communication and Processing Units)

M2M Devices Domain

Communication Gateway

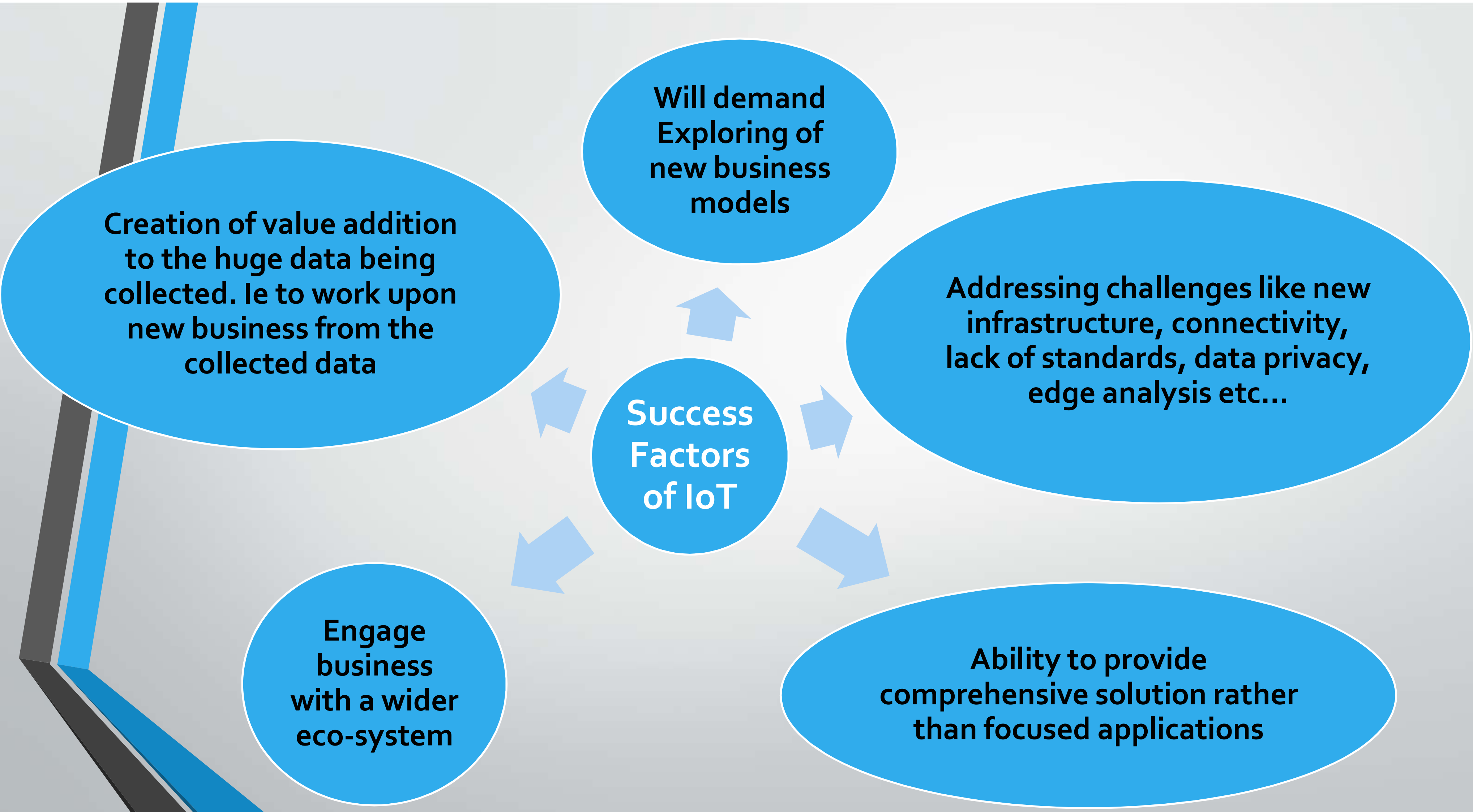
Connectivity Interface (Communication and Processing Units) and Edge Computing (Data Element Analysis and Transformation)

Physical Devices and Controllers (the Things in IoT) [Sensors, Machines, Devices, Intelligent Edge Nodes of Different Types]

Connect
Collect
Assemble
Analyse

M2M Software development tools

- Mango – Open source M2M web based software
- Mainspring from M2M labs
- DeviceHive is an M2M communication framework – enables connecting devices to IoT – web based management



Success Factors of IoT

Will demand Exploring of new business models

Addressing challenges like new infrastructure, connectivity, lack of standards, data privacy, edge analysis etc…

Ability to provide comprehensive solution rather than focused applications

Engage business with a wider eco-system

Creation of value addition to the huge data being collected. Ie to work upon new business from the collected data

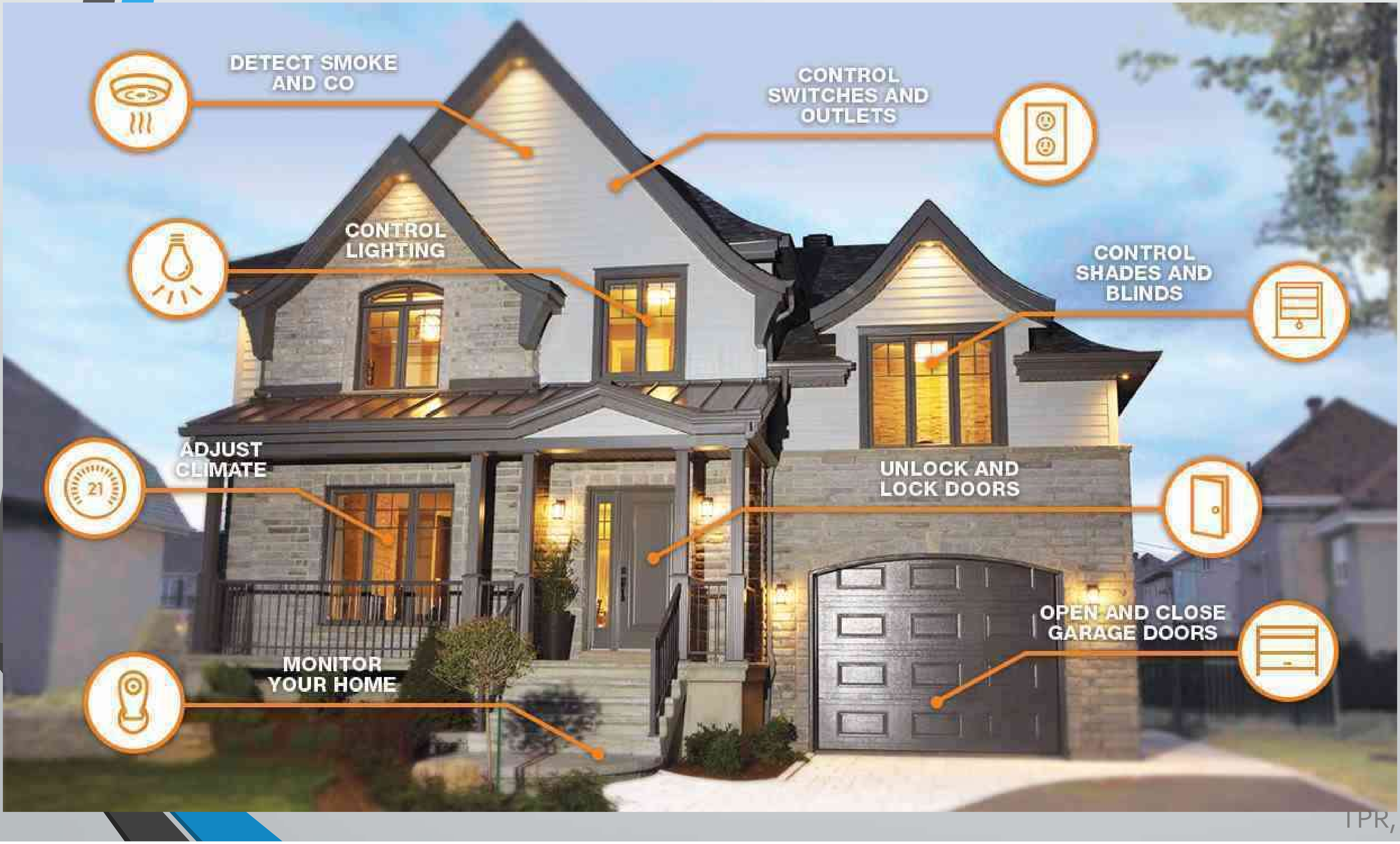
Examples of IoT
/
Application Areas

Features of smart watches

- Ability to make phone call
- WiFi & Bluetooth
- GPS enabled
- Health applications & UV monitors
- Tracks health parameters all the time
- Enables payment using wallet
- Video chat
- News & social networking
- Navigation
- Gyroscope, Accelerometer, heart sensor, UV sensor, skin temp sensor, barometer, light sensor etc…

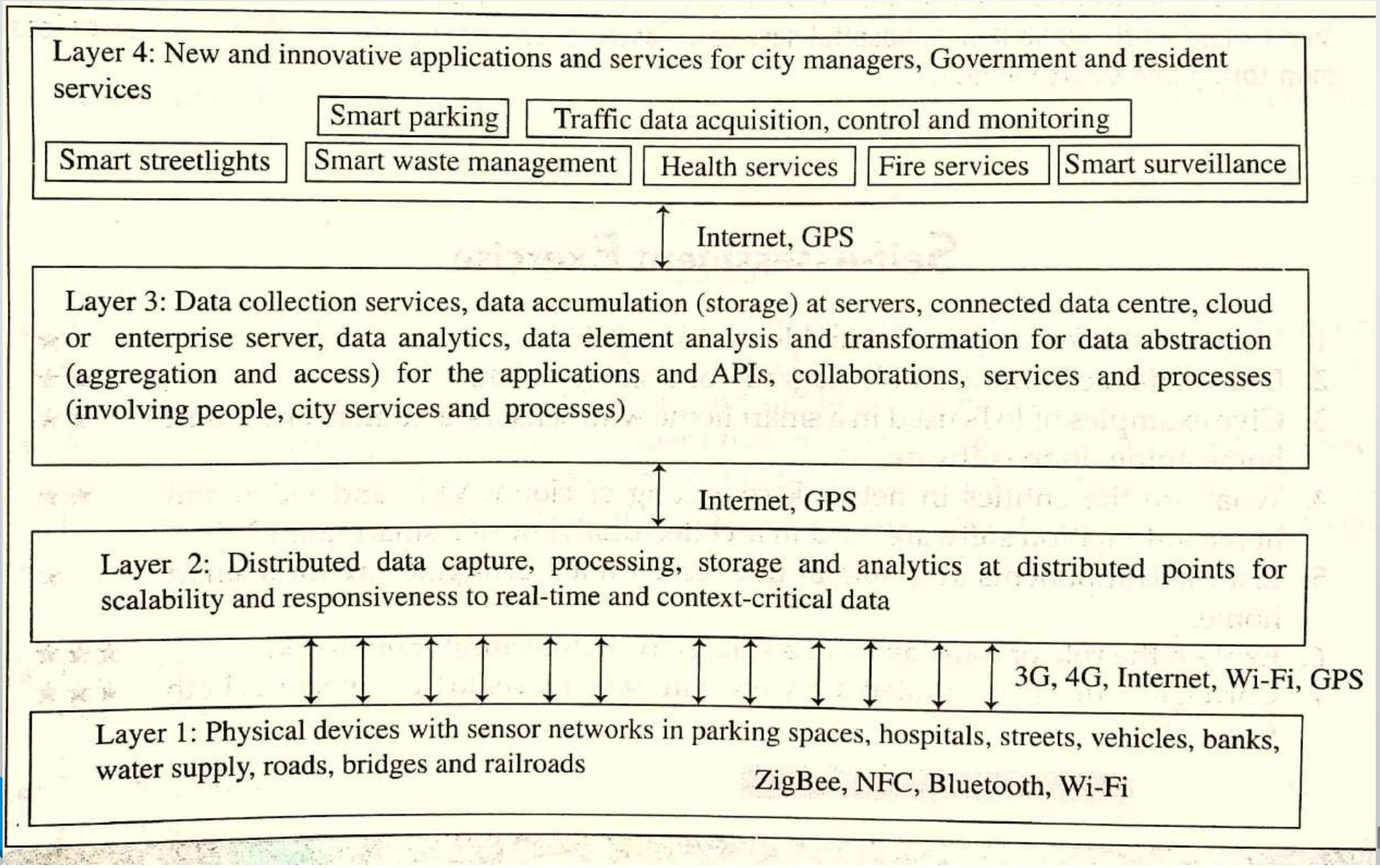
Smart homes – A home with sensors and actuators, connected and managed via internet

- Cameras, security sensors, thermostats, smart plugs, light and entertainment systems, smoke detector, energy meter interface (electricity, gas, water), surveillance cameras, speakers, LED lights etc…
- Home automation softwares:
  - Intel based intelligent gateway
  - OpenHAB – An Eclipse IoT based project – runs on java enabled system
  - The Thing System. Language is "Node.js". Can fit into a raspberry pi


DETECT SMOKE AND CO
CONTROL SWITCHES AND OUTLETS
CONTROL LIGHTING
CONTROL SHADES AND BLINDS
21
ADJUST CLIMATE
UNLOCK AND LOCK DOORS
OPEN AND CLOSE GARAGE DOORS
MONITOR YOUR HOME

TPR, MACE

Smart Cities – an architectural framework by CISCO



Layer 4: New and innovative applications and services for city managers, Government and resident services

Smart parking

Traffic data acquisition, control and monitoring

Smart streetlights

Smart waste management

Health services

Fire services

Smart surveillance

Internet, GPS

Layer 3: Data collection services, data accumulation (storage) at servers, connected data centre, cloud or enterprise server, data analytics, data element analysis and transformation for data abstraction (aggregation and access) for the applications and APIs, collaborations, services and processes (involving people, city services and processes)

Internet, GPS

Layer 2: Distributed data capture, processing, storage and analytics at distributed points for scalability and responsiveness to real-time and context-critical data

3G, 4G, Internet, Wi-Fi, GPS

Layer 1: Physical devices with sensor networks in parking spaces, hospitals, streets, vehicles, banks, water supply, roads, bridges and railroads

ZigBee, NFC, Bluetooth, Wi-Fi

DESIGN PRINCIPLES FOR CONNECTED DEVICES

- There must be a specified protocol (rule) followed at each level (layer) of **data** transfer between connected devices.
- For IoT / M2M, there should be some principles for **data** transfer

Layer

- A stage during a set of actions

Physical layer

- Refers to a layer at transmitting node / receiving node for data bits. It's the lowest layer and uses physical systems for transmission like WiFi, LAN etc…

Application layer

- Layer for transmitting/receiving data of an application

Domain

- A set of softwares having specific applications/capabilities.

Gateway

- Software for connecting two application layers – one at sender and other at receiver

IP – Internet Protocol

- IPv6 or IPv4

Header

- Set of octets containing information about data being sent. (The **octet** is a unit of digital information in computing and telecommunications that consists of eight bits.)

Packet

- Packaged data stack which routes over the network. Packet size has limitations. In IPv4, packet size limit is 2^{16} B (2^{14} words where one word = 4 Bytes = 4 octet)

Protocol Data Unit (PDU)

- A unit of data which is specified in a protocol of a given layer

Maximum Transmission Unit (MTU)

- Largest size frame or packet or segment specified in octets (1 octet = 1 byte) that can be sent in a packet

Star N/W

- Number of nodes interacting with a master node

Mesh node

- A number of nodes that interconnects with each other

Master

- The one who initiates pairing with devices in a star topology

Slave

- One that pairs with master, uses clock signals from master for synchronization and uses address assigned by the master at beginning

Router

- A device capable of storing paths to each destination to which, it has logical links

ISM Band

- Industrial, Scientific, Medical radio frequency bands - 2.4GHz & 433MHz (Asia)

RECENT INITIATIVES OF INTERNATIONAL ORGANISATIONS FOR DESIGN STANDARDIZATION OF IOT/M2M

IETF – Internet Engineering Task Force

- Suggests specifications of layers and engg aspects of IoT

ITU-T – International Telecommunication Union for Telecommunication

- Suggested a reference model for IoT domain

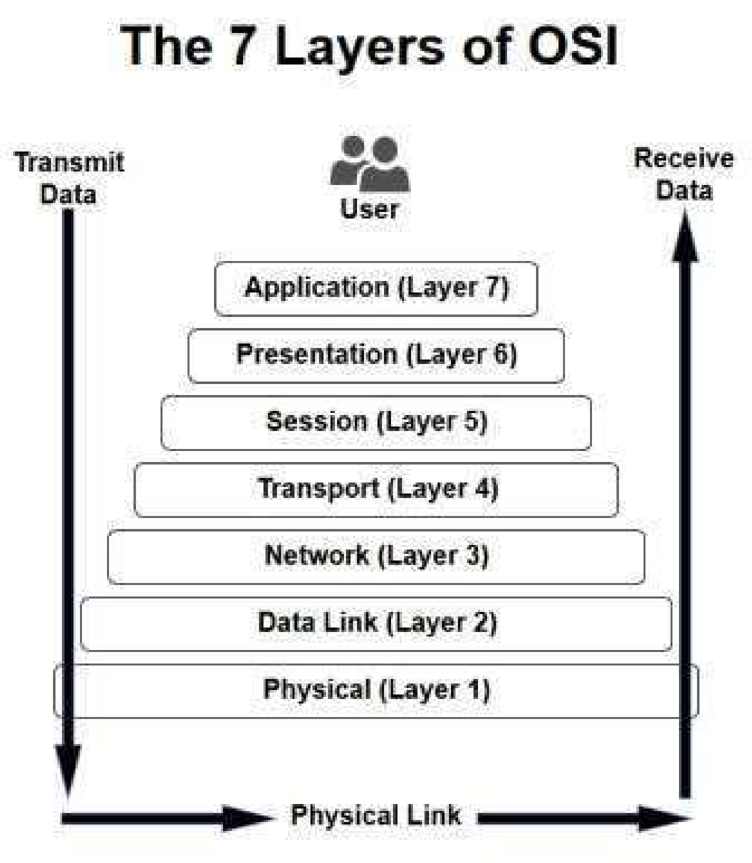
ETSI – European Telecommunication Standards Institute

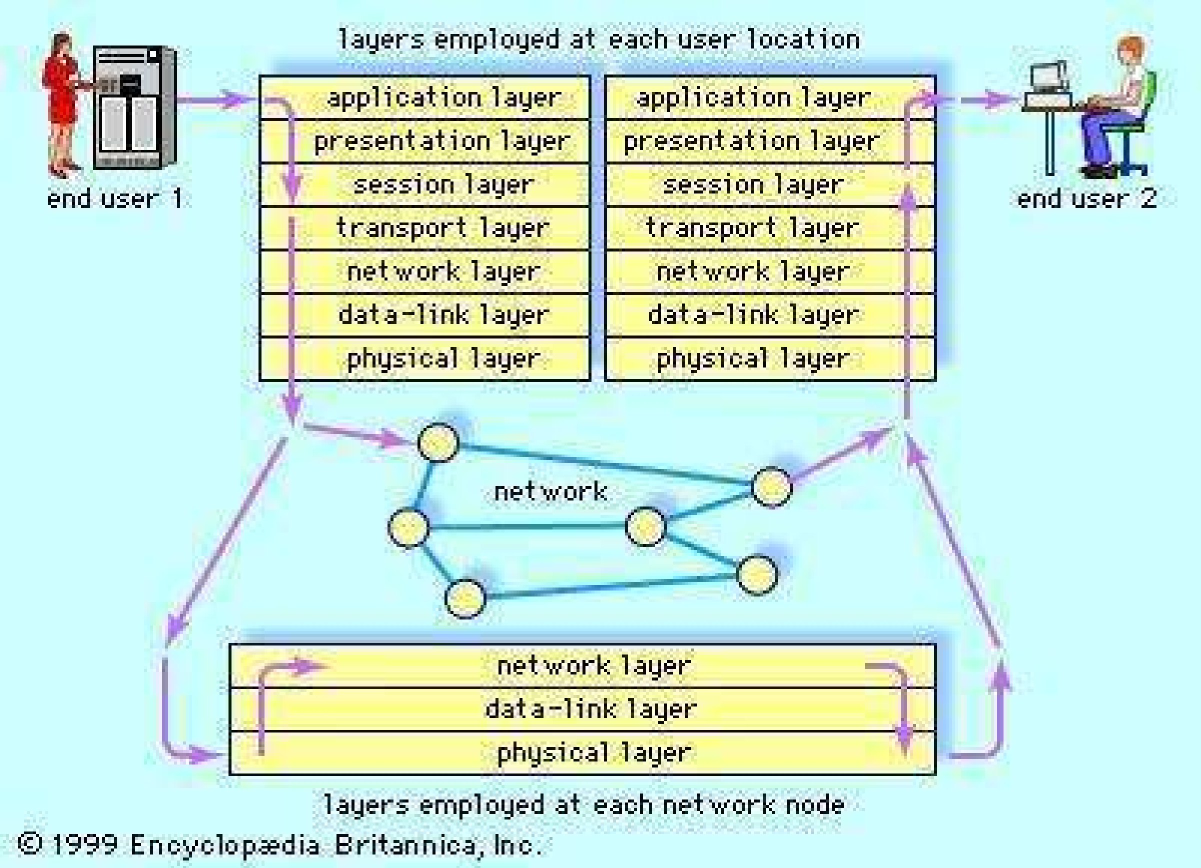
- Developed a set of standards for the n/w, devices and gateway domains in M2M

OGC – Open Geospacial Consortium

- Suggested open standards for sensors' discovery, capabilities quality and other aspects with support to geographical information web support

- The Open Systems Interconnection **(OSI)** model is a conceptual model that characterizes and standardizes the communication functions of a telecommunication or computing system without regard to its underlying internal structure and technology. OSI protocol means a family of information exchange standards jointly developed by ISO and ITU-T (International Telecommunication Union for Telecommunications) . The following 7 layered model is a standard model – gives a basic outline for designing a communication n/w.





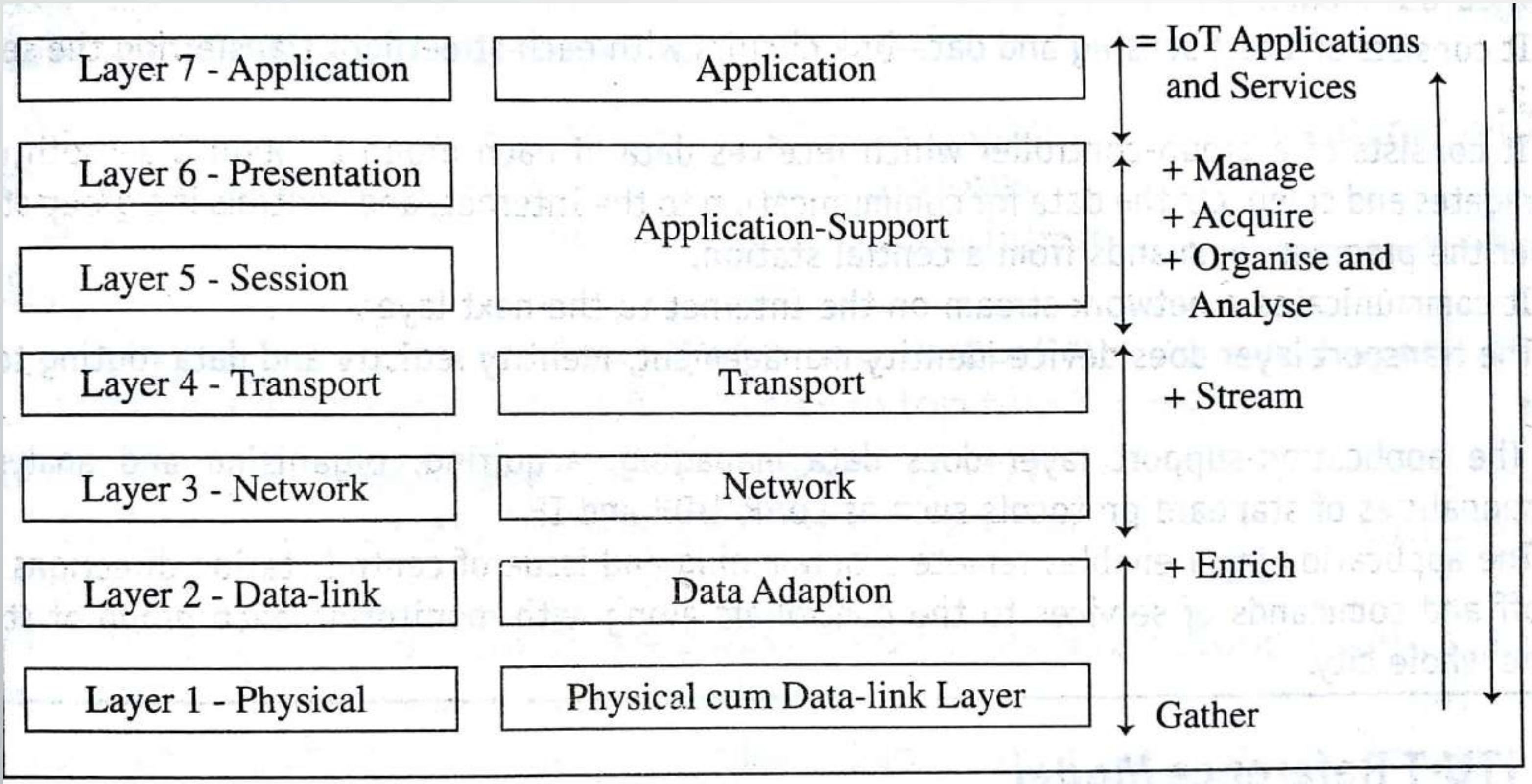
© 1999 Encyclopædia Britannica, Inc.

CE

6 levelled Modified OSI model for IoT/M2M - by IETF (Internet Engineering Task Force)

Modifications proposed are shown in the middle

The basic architecture equation of IoT application/service= Gather + Enrich + Stream + Manage + Acquire+Organise+Analyse (7 levelled)



.1 Seven-layer generalised OSI model (on left) and IETF six layer modified OSI model for IoT/M2M (in the middle), and similarity with the conceptual framework Equation 1.2 (on right) for IoT applications and services

Architectural layers using modified OSI model for Internet of smart street lights application

Level 1

- Smart sensing and data link circuits with each streetlight transferring the sensed data to level 2

Level 2

- Has a group controller which receives data of each group through Bluetooth/zigbee
- Aggregates & compacts data for communication to internet& controls the group streetlights as per program command from central station

Level 3

- Communicates a n/w stream on internet to the next layer

Level 4

- Transport layer does device identity management, identity registry and data routing to the next layer

Level 5

- Application support layer does data managing, acquiring, organizing and analyzing using protocols like CoAP, UDP, IP

Level 6

- Application layer enables remote programming and issue of central station directions to switch on/off and commands of services to the controllers along with monitoring each group of streetlights in the whole city

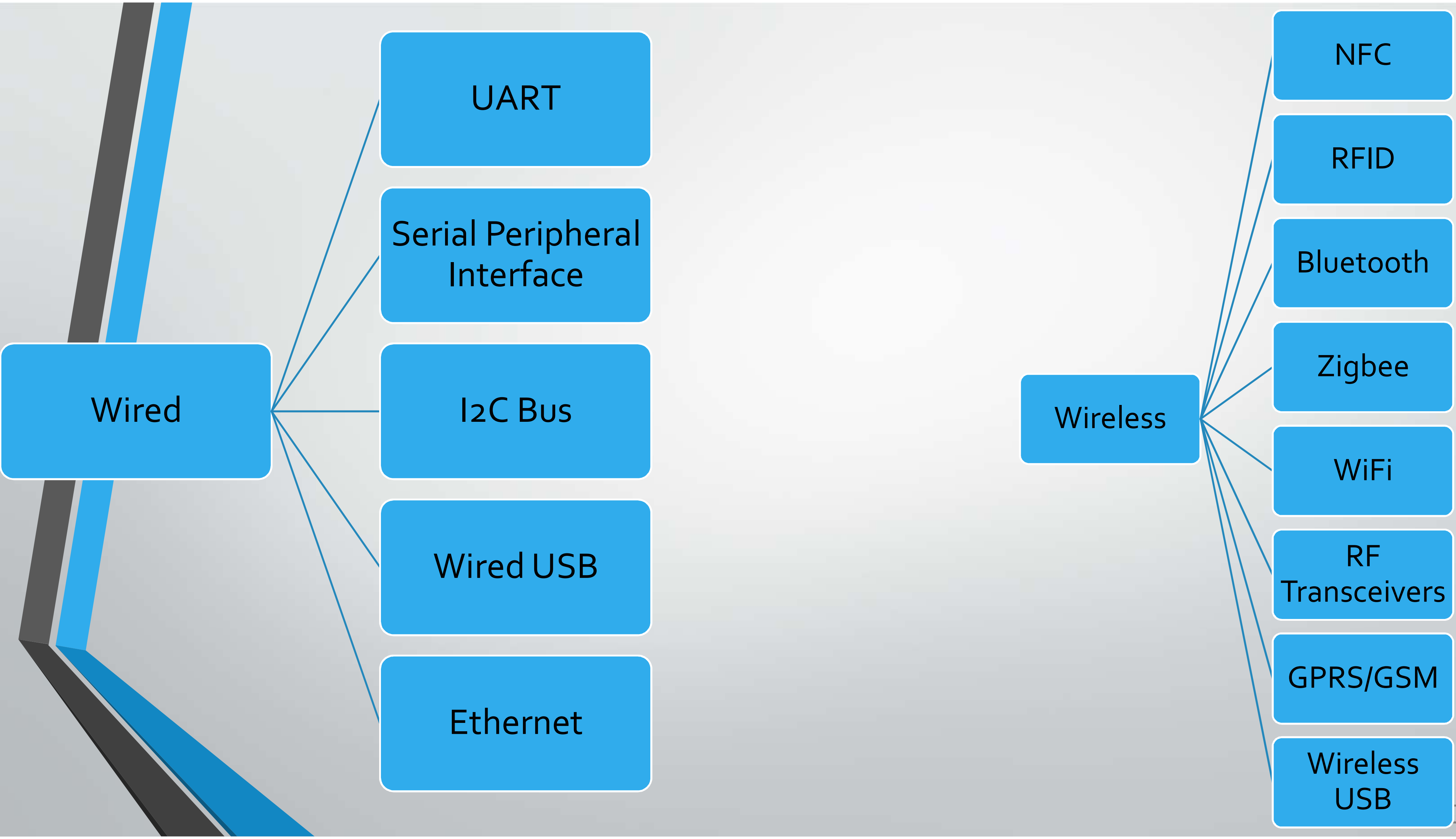
ITU-T REF MODEL

ETSI M2M DOMAIN

COMMUNICATION TECHNOLOGIES

Will be dealing with wireless and wired communication technologies for physical and data-link layer functions

- Physical and data-link layer consists of a local area network (LAN) or personal network
- The above can be realized using Wireless or Wired technologies



Wired

UART

Serial Peripheral Interface

I2C Bus

Wired USB

Ethernet

Wireless

NFC

RFID

Bluetooth

Zigbee

WiFi

RF Transceivers

GPRS/GSM

Wireless USB

Near Field Communication

- Enhancement of standard ISO/IEC^2 14443 standard for contactless proximity card
- short distance 10– 20cm
- eg: proximity card reader, RFID, IoT device, mobile payment wallet, electronic keys for car, house, office, biometric passport reader…
- NFC devices transmit and receive at the same instant
- Setup time very less ~ 0.1 sec
- Device data transfer rates106kbps, 212kbps, 424kbps and 848 kbps (kilo bits per sec)

3 Modes of NFC

- Point to Point(P2P) – both devices use active devices in which, RF fields alternatively generate when communicating. Eg: mobile to mobile
- Card emulation mode – communication without interruption for read and write. Eg in smart card + smart card reader. Standards used – FeliCa and MiFare
- Reader mode – Using NFC device to read passive RFID device. The RF field is generated by an active NFC device. This enables passive device to communicate

RFID – Radio Freq Identification

- Automatic identification
- Uses internet
- RFID device functions as tag / label

Bluetooth BR/EDR & Bluetooth Low Energy

- IEEE 802.15.1 standard protocol
- Devices form a WPAN
- Two modes:
  - Bluetooth BR/EDR – Basic Rate 1Mbps / Enhanced Data Rate 2Mbps & 3Mbps
  - Bluetooth Low Energy / smart Bluetooth – 150m @ 10mW – 1Mbps
- v5 – June 2016 capacity increased by 800%, range by 4 times, speed by 2 times
- Auto-synchronization, n/w self discovery/configuration/healing
- Smaller packets in LE mode

Zigbee

- IEEE 802.15.4
- Devices form a WPAN of sensors, actuators, appliances, controllers etc…
- IEEE 802.15.4-conformant devices may use one of three possible frequency bands for operation (868/915/2450 MHz), 2.4GHz is the globally accepted frequency.

Zigbee IP - is the first open standard for an IPv6-based full wireless mesh networking solution and provides seamless Internet connections to control low-power, low-cost devices. It connects dozens of the different devices into a single control network.

- Layer 1 Protocol Data Unit (PDU) = 127B
- Devices can function in 6 modes:
  - > end point > zigbee-zigbee devices router > zigbee n/w coordinator > zigbee IP coordinator > zigbee IP router > Zigbee IP host
- Capable of IPv6 connectivity
- Application in big scale automation & remote controls, smart metering
- Self configuring / healing
- Range: 10 – 200m @ 250kbps, low power operation

Wi-Fi

- A trademark phrase meaning IEEE 802.11 protocol
- Enables wireless Local Area N/W (WLAN)
- Connects enterprises, universities, offices using the internet
- Almost all smart devices have Wi-Fi interface
- 2.4Ghz IEEE 802.11b adaptor or 5GHz IEEE 802.11a or 802.11g or 802.11n
- Easy installation, simplicity & flexibility
- 30m to 125m
- Limited coverage areas – a room – 802.11a
- Wider coverage – 802.11b – 11Mbps within 30m
- 802.11g for high data rates – 54Mbps
- 802.11n for very high data rates – 600Mbps
- Provides dynamic environment of n/w expandability & scalability
- Provides Security, integrity & reliability

RF Transceivers & RF Modules

- Transceiver = transmitter+receiver
- IoT/M2M enploys ISM band RF modules with transceivers
- Security, telemetry, home automation, fleeting, healthcare, automotive industry, banking…
  - > RF interface/physical layer consisting of sensors, actuators, controllers and gateways, transceivers…
  - > RF n/w architecture includes overall system , servers, managing the RF signals during sleep mode and active mode.

GSM/GPRS

- General Packet Radio Services
- Global System for Mobile communication
- IoT devices can use / access wireless networks through GPRS cellular network

Wireless USB

- Wireless extension of usb 2.0 and operates at UWB 5.1GHz to 10.6GHz

Wired Communication

UART/USART Serial Comm

- Universal Asynchronous Transmitter – serial communication of 8 bits serially with a start bit on the Transmission Data Output line (TxD)
- Asynchronous – means all bytes in a frame transmit- resulting in phase differences b/w successive bytes. The cock information does not transmit along with data. Receiver clock also does not synchronize with data
- USART – Universal Synchronous Asynchronous Transmitter enables data transmission in both synchronous & asynchronous modes.

Serial Peripheral Interphase (SPI)

- Serial synchronous communication method
- Master – source of synchronous o/p
- Slave – Receiver of serial synchronous data & clock information from master

I2C Bus

- Inter Integrated Circuit
- IC's mutually network through a common synchronous serial bus
- 4 modes – master transmit, master receive, slave transmit, slave receive
- Developed by Philips semiconductors
- I2C bus has 2 lines – one for clock and other for bidirectional data

Wired USB

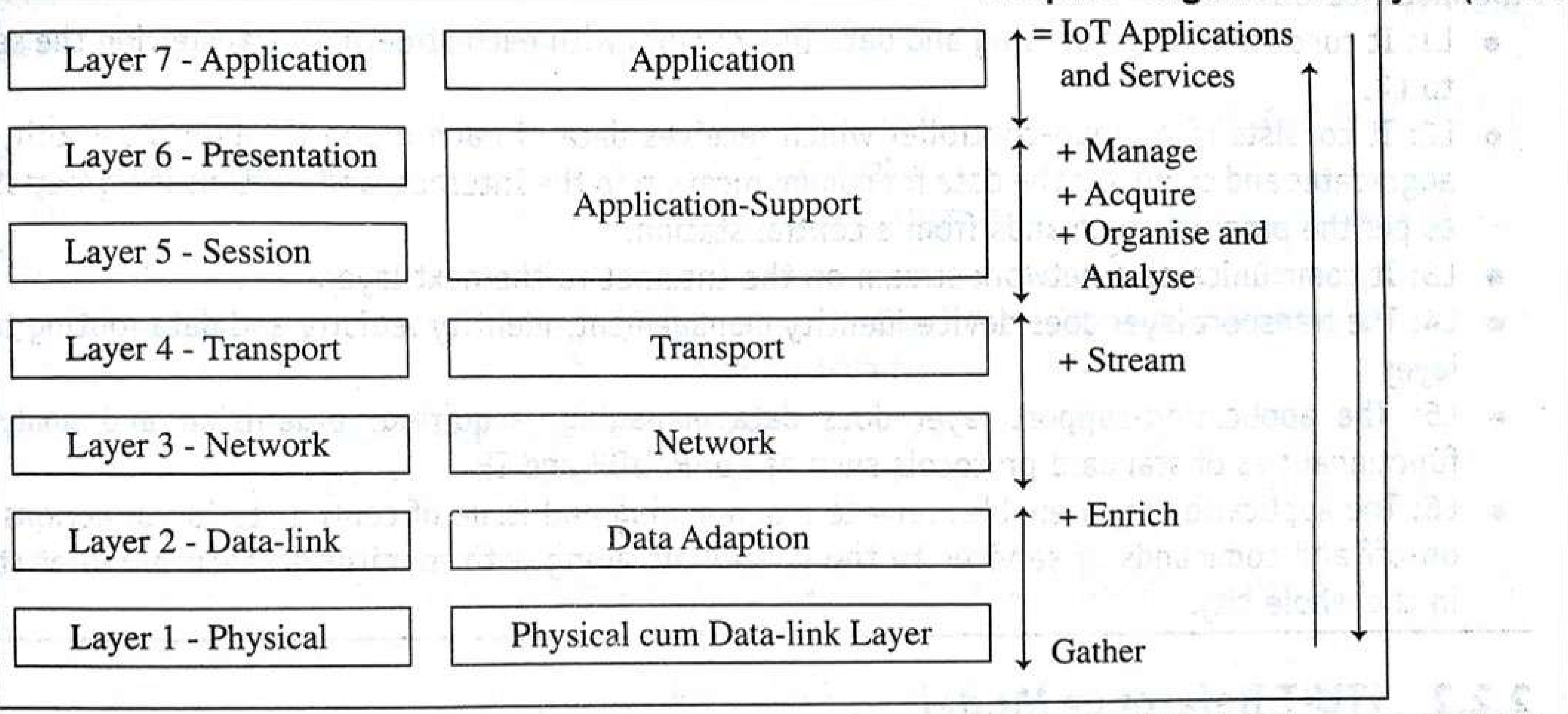
- Universal Serial Bus
- Connecting keyboard, printer or scanner
- USB is a bus b/w host system and a number of interconnected peripheral devices (max 127)
- Upto 12Mbps
- 3 standards – USB 1.1 (1.5 & 12Mbps), USB 2.0 (480Mbps), USB 3.0 (5Gbps), USB 3.1 (10Gbps)

DATA MANAGEMENT

6 levelled Modified OSI model for IoT/M2M - by IETF

Modifications proposed are shown in the middle

The basic architecture equation of IoT application/service= Gather + Enrich + Stream + Manage + Acquire+Organise+Analyse (7 levelled)



Layer 7 - Application | Application | = IoT Applications and Services

Layer 6 - Presentation | Application-Support | + Manage + Acquire + Organise and Analyse

Layer 5 - Session | Application-Support

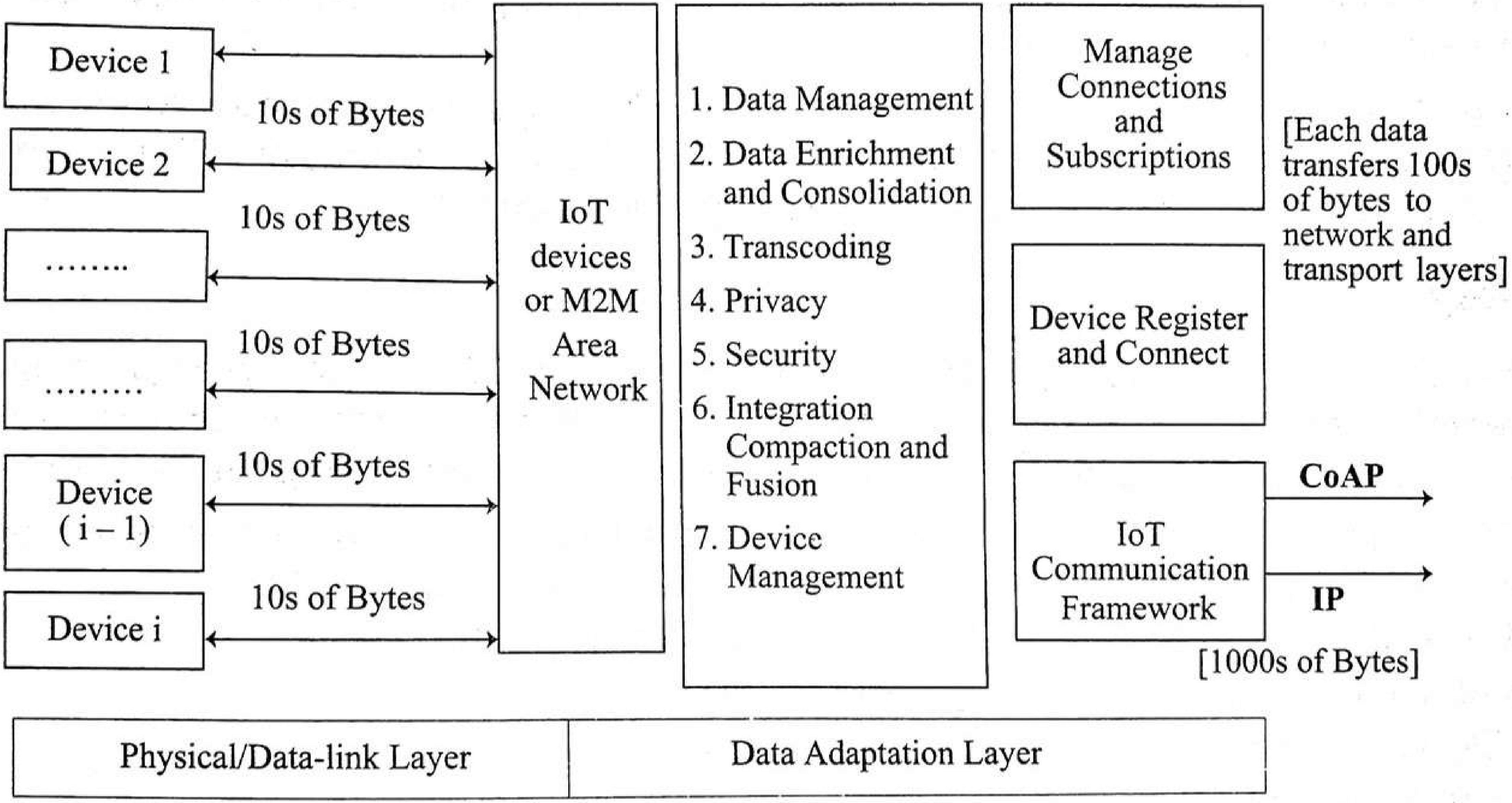
Layer 4 - Transport | Transport | + Stream

Layer 3 - Network | Network

Layer 2 - Data-link | Data Adaption | + Enrich

Layer 1 - Physical | Physical cum Data-link Layer | Gather

.1 Seven-layer generalised OSI model (on left) and IETF six layer modified OSI model for IoT/ M2M (in the middle), and similarity with the conceptual framework Equation 1.2 (on right) for IoT applications and services



Device 1

Device 2

……..

………

Device (i – 1)

Device i

10s of Bytes

10s of Bytes

10s of Bytes

10s of Bytes

10s of Bytes

IoT devices or M2M Area Network

1. Data Management
2. Data Enrichment and Consolidation
3. Transcoding
4. Privacy
5. Security
6. Integration Compaction and Fusion
7. Device Management

Manage Connections and Subscriptions

Device Register and Connect

IoT Communication Framework

[Each data transfers 100s of bytes to network and transport layers]

CoAP

IP

[1000s of Bytes]

Physical/Data-link Layer

Data Adaptation Layer

Gateway

- Gateway located at data adaptation layer provide functions like data privacy, security, enrichment, consolidation , transformation and device management.

Gateways have basically two purposes

- 1. Data Management & Consolidation
- 2. Connected Device Management

1. Data Management and consolidation gateway has the following functions:

Transcoding

- Data Adaptation, conversion, change of protocol, format/code using software. Renders response and messages acceptable in IoT.
- Similarly, IoT device requests are adapted, converted and changed into required format acceptable by servers.
- Transcoding also involves compression/decompression.

Privacy

- Some data like medical, change of inventories etc..need privacy
- Ensure that data at the receiving end is considered anonymous
- Identity management, Authentication, Authorisation, Trust, Reputation are the components of privacy

Secure data Access

- Authentication & authorization needed.
- End to end security - uses security protocol at each layer, physical, logical & transport layers during communication

Data Gathering and enrichment

- Acquisition, validation, storage, processing, retention, analysis
- Data can be gathered via:
  - > Polling – data sought by addressing a device
  - > Event based – data sought from a device upon an event.
  - > Scheduled – Data sought from a device at selected intervals.
  - > Continuous monitoring – accessing data continuously
- Data enrichment is about adding value, security, usability etc…

Data dissemination (spreading)

- Aggregation – joining together present and previous data after removing duplicate entries
- Compaction – making information short without affecting meaning
- Fusion – formatting the data in parts, from different types of sources

Energy consumption during data dissemination

- Limited Battery life
- Complex computations/transformations consume much energy
- Higher data rate
- Higher frequency band
- Higher sampling rate

Data Source and destination

- ID's - each device resource is assigned an ID
- Address - destination address
- Eg: 32 bit IPv4 address, 128 bit IPv6 address…

Data Types/ Structures

- Temporary data
- Spatial data (depends on location)
- Real time data
- Real world data
- Proprietary data
- Big Data - unstructured voluminous

2. Device Management Gateway

- Managing the connected devices
- DM – Device Management means provisioning for the device ID/address distinguishing it from other resources
- DM Server – server for assigning device ID/address, activating, configuring the devices
- Gateway does the following tasks connected with device management:
  - > Does forwarding function of data
  - > Does protocol conversion
  - > Does proxy functions in lossy environments

Design Principles for Web Connectivity

Web Connectivity

- A communication gateway will enable web connectivity for a system.
- But for an IoT platform, specific protocols and methods enable web connectivity for a connected device network

App

- Application softwares like measuring and sending the temperature data of an area at specified time intervals

Application Programming Interface (API)

- Refers to a software component which receives messages from one end like an app/client/input
- An API may get i/p to or from a server
- It can consist of GUI's or program sequences which enable easier development of an application

Web Service

- A service available over internet using web protocols. Eg: weather reports, traffic density etc...

Object

- Collection of resources. Eg: "Time_Date" object having provisions to collect the day, date, time, minute, seconds etc...

Object Instance

- For a "Time_Date" object, object instance example is "Birth_Date"

Communication Gateway

- Functions as communication protocol translator to help communication. Eg: Zigbee & IP Networks

Client

- Software object which makes a request (or an API associated)

Server

- A software which responds to a request. Also sends alerts, notifications. It has access to resources, databases, objects. A server can function in a separate computer system or on an internet connected system

Web Object

- One that retrieves information using Web protocol

Broker

- An object which arranges communication between two ends, like server and application

Communication Protocol

- Defines the rules and conventions for communication b/w networked devices & b/w systems. It includes how devices communicates and how data is packaged and sent

Web Protocol

- Defines rules and conventions for communication b/w web server and web client, web objects

Firewall

- Protects the server from unauthentic resources

Header - A set of words containing information about processing at a particular layer

State - Ref to an aspect about data or object received

Resource - Something that can be read, written or executed

Resource Instance - It has a resource ID - Resource identifier can be a path specification like URL (universal resource locator) or URI(universal resource identifier)

TCP - Transmission control protocol ; IP - Internet Protocol

URI - **Universal Resource Identifier** - used for saved resources like contacts or address books. It is identifier for a specified resource. Like a page, book, document etc…

URL - **Universal Resource Locator -** Used for retrieving a resource by a client. The resource may be located at a remote server. Eg: http://www.mace.ac.in Can be compared to. If URI is my name, URL is my address

Datagram - Limited size data (2^{16} bytes). A **datagram** is an independent, self-contained message sent over the network whose arrival, arrival time, and content are not guaranteed.

Packet - Refers to data - When we use connection-oriented protocol for communication, packets are used

REST - Representational State Transfer

- Software architectural style - referring to ways of defining identifiers for the resources, methods, access methods and data transfer during interactions
- It also specifies the practices, constraints, characteristic and guidelines for creating a scalable (can be developed according to size) web services.
- The goal of REST is to increase performance, scalability, simplicity, modifiability, visibility, portability, and reliability.

RESTful - One which follows REST constraints ; REST API's - An API that uses HTTP requests to get / put data

UDP - User Datagram Protocol

- Is a protocol at the transport layer for the web using internet and constrained RESTful environment
- Unlike TCP, UDP is a fire and forget style protocol.
- It specifies ways of enveloping datagram by header words

HTTP - Hyper Text Transfer Protocol

- An application layer protocol for use of hypertext as app data transfer protocol

Hypertext - Text embedded with hyperlinks

Hyperlink-Specification of the URL

Browser

- Client software which displays hypertext that enables navigation to hypertext links

Hyperlink

- A specification of the URL for a resource path, so that a link can be established b/w two objects

HTML – Hypertext Markup Language

- Language for creating a hypertext which refers to text that embeds text, images, audio, and video, list, tables etc…

XML – Extensible Markup Language

- Language which enables creation, sending and receiving documents, messages, commands, query responses, and creation of forms.

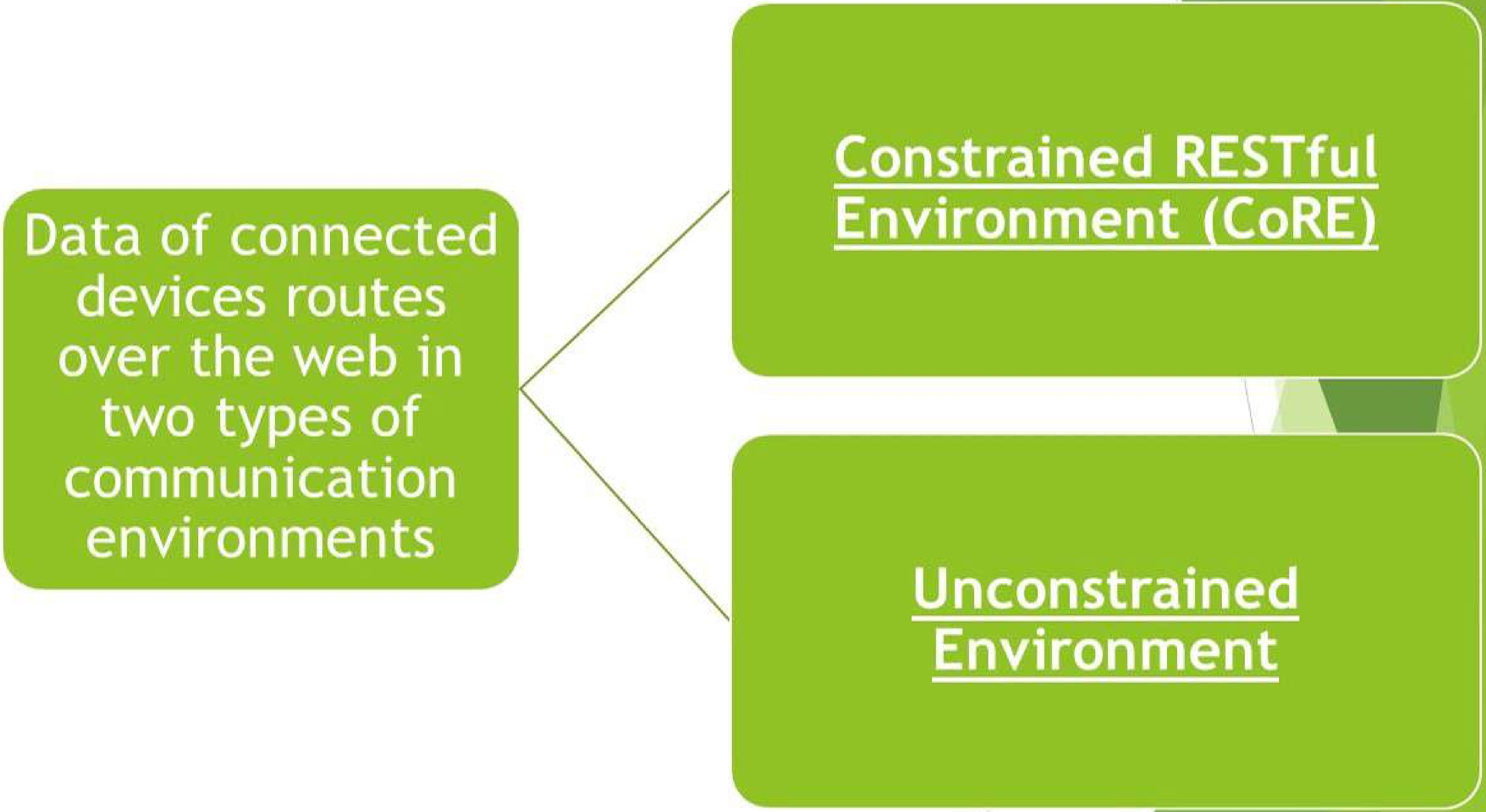
Browser

- Client software which displays hypertext that enables navigation to hypertext link shoen on screen

Framework

- Refers to provisions for a number of software libraries and API's

Web Communication Protocols for Connected Devices



Data of connected devices routes over the web in two types of communication environments

Constrained RESTful Environment (CoRE)

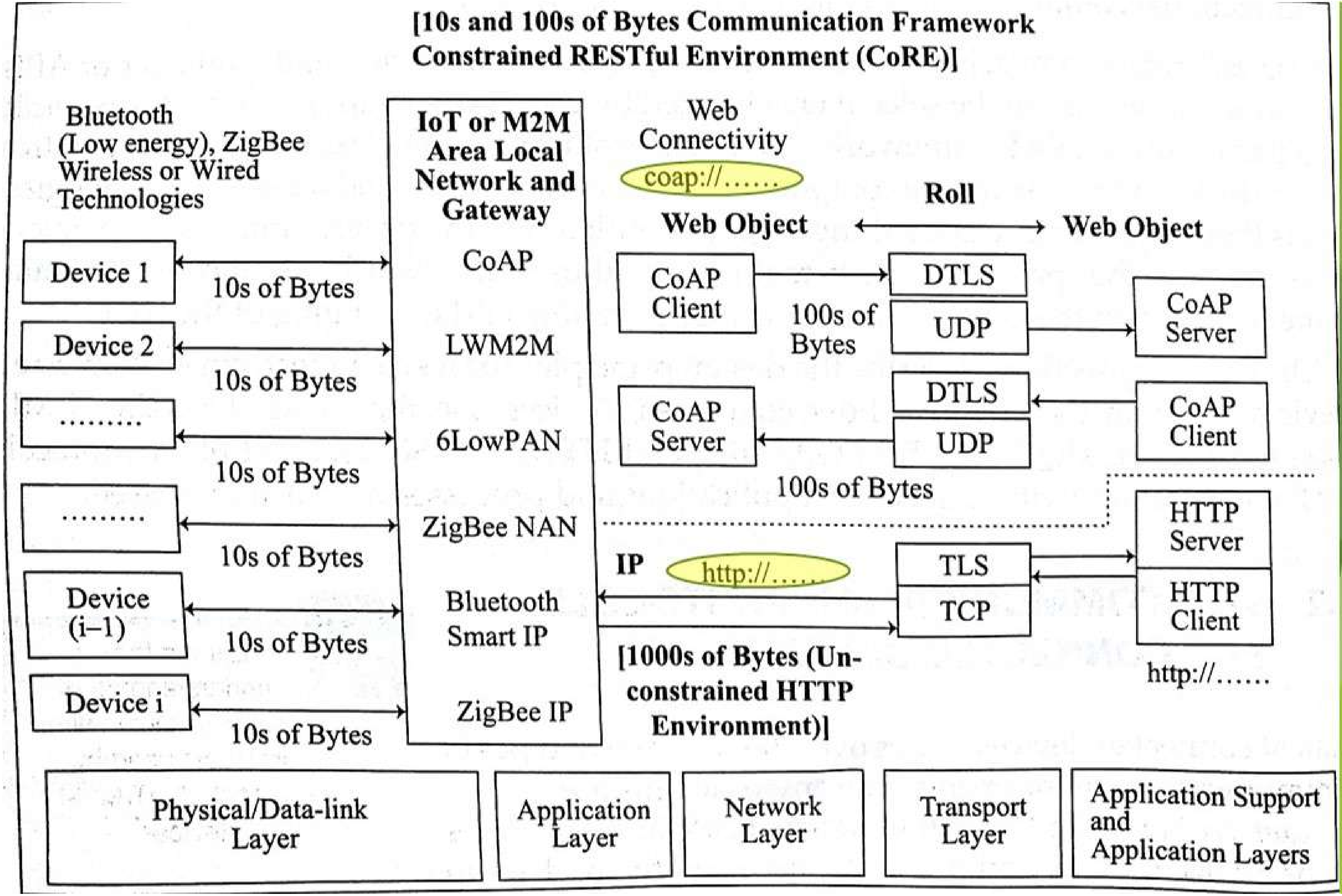
Unconstrained Environment

Constrained RESTful Environment (CoRE)

- Uses a version of REST protocol
- Devices are constraint - their data is limited in compared to data exchanges in HTTP, TCP and IP. Typically, the device sends/receives only 10s of bytes of data. After enriching, it becomes 100s of bytes.
- Data routing is also constrained - when **Routing Over a n/w of Low power and data Loss (ROLL).** ROLL networks have low power transceiver
- Another constraint is that devices sleep most of the time in low power environment & awaken only when required.
- Devices' connectivity may make/break many times, and have limited data size

Unconstrained Environment

- Web applications use HTTP, and RESTful HTTP for web client and web server communication. A web object consist of 1000s of bytes. Data routes over IP networks. Applications and services use IP and TCP (transmission control protocol)



[10s and 100s of Bytes Communication Framework Constrained RESTful Environment (CoRE)]

Bluetooth (Low energy), ZigBee Wireless or Wired Technologies

Device 1

Device 2

....

....

Device (i–1)

Device i

10s of Bytes

10s of Bytes

10s of Bytes

10s of Bytes

10s of Bytes

10s of Bytes

IoT or M2M Area Local Network and Gateway

CoAP

LWM2M

6LowPAN

ZigBee NAN

Bluetooth Smart IP

ZigBee IP

Web Connectivity

coap://……

Roll

Web Object

Web Object

CoAP Client

DTLS

UDP

100s of Bytes

CoAP Server

CoAP Server

DTLS

UDP

CoAP Client

100s of Bytes

IP

http://……

TLS

TCP

HTTP Server

HTTP Client

http://……

[1000s of Bytes (Un-constrained HTTP Environment)]

Physical/Data-link Layer

Application Layer

Network Layer

Transport Layer

Application Support and Application Layers

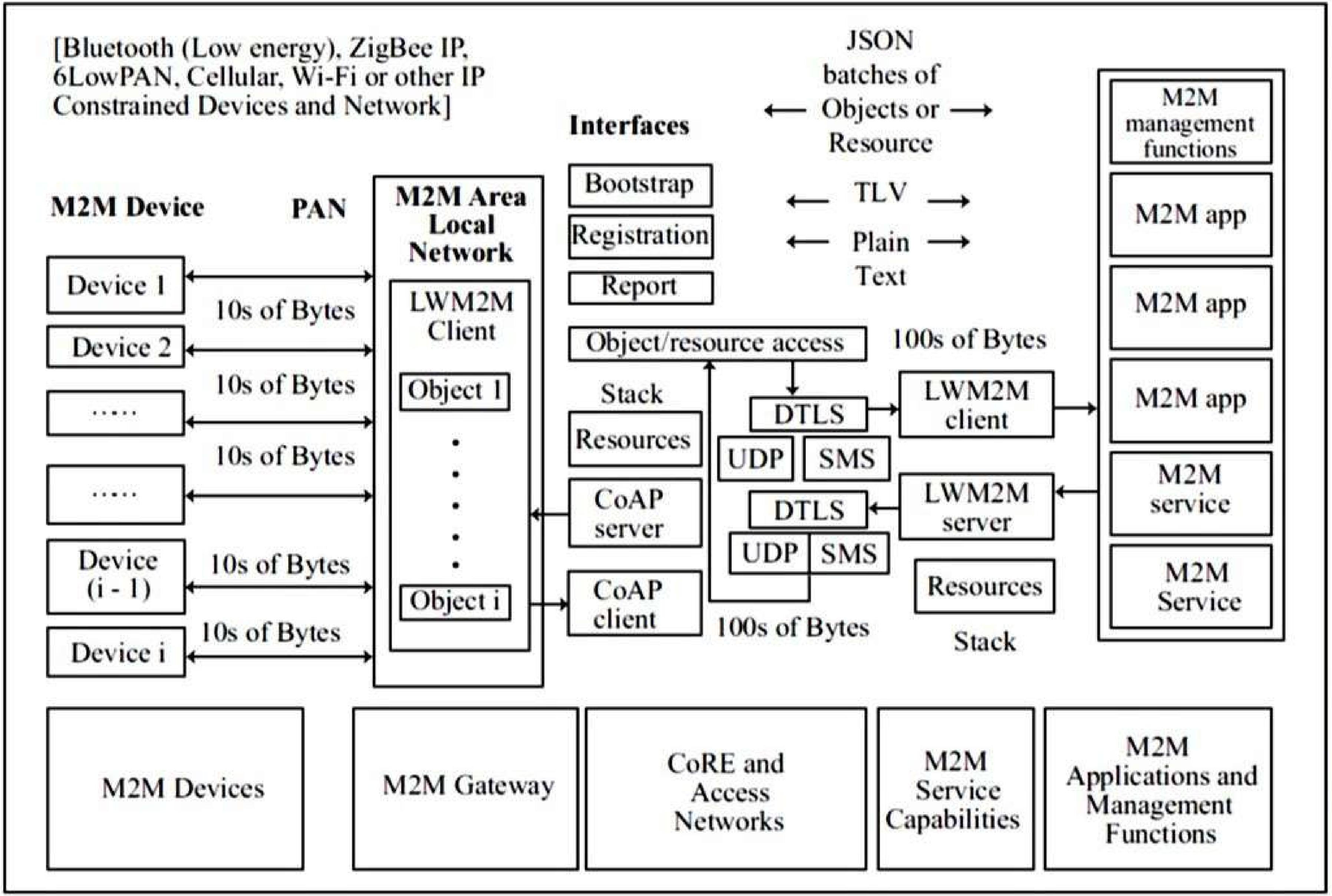
IoT device or machine applications need constrained environment protocols such as CoAP and LWM2M

Constrained Application Protocol

- IETF recommends COAP, which is CoRE using ROLL data network
- CoAP web objects communicate using request/response interaction model
- Uses object model for resources & each object can have multiple instances
- Supports resource directory
- Resource identifiers use the URI – coap://... instead of http://...
- CoRE communication is asynchronous communication over ROLL
- Integrates easily with the web using CoAP cross protocol proxies. This is facilitated since HTTP and CoAP both share REST model.

Light Weight Machine to Machine Communication Protocol (LWM2M)

- Its an application layer protocol specified by Open Mobile Alliance(OMA) for transfer of service data/messages.
- Used in M2M
- Enables functionalities for device management in cellular/ sensor networks
- Light weight – means data transfer format b/w client and server are binary and has Tag Length Value (TLV) or JavaScript Object Notation(JSON) & transfers upto 100s of bytes.
- This protocol enables communication b/w LWM2M client at IoT device and an LWM2M server at the M2M application and service capability layer.



M2M devices local area network connectivity, and constrained devices network connectivity with M2M applications and services using LWM2M OMA standard specifications of LWM2M

MESSAGE COMMUNICATION PROTOCOLS

CoAP-SMS and CoAP-MQ

- **CoAP-SMS** is a protocol when CoAP object uses IP and cellular networks for sending SMS
- URI used is coap+sms:// instead of coap://
- SMS consist of 160 characters
- **CoAP-MQ** is a message que protocol using a broker and RD(Resource Directory)

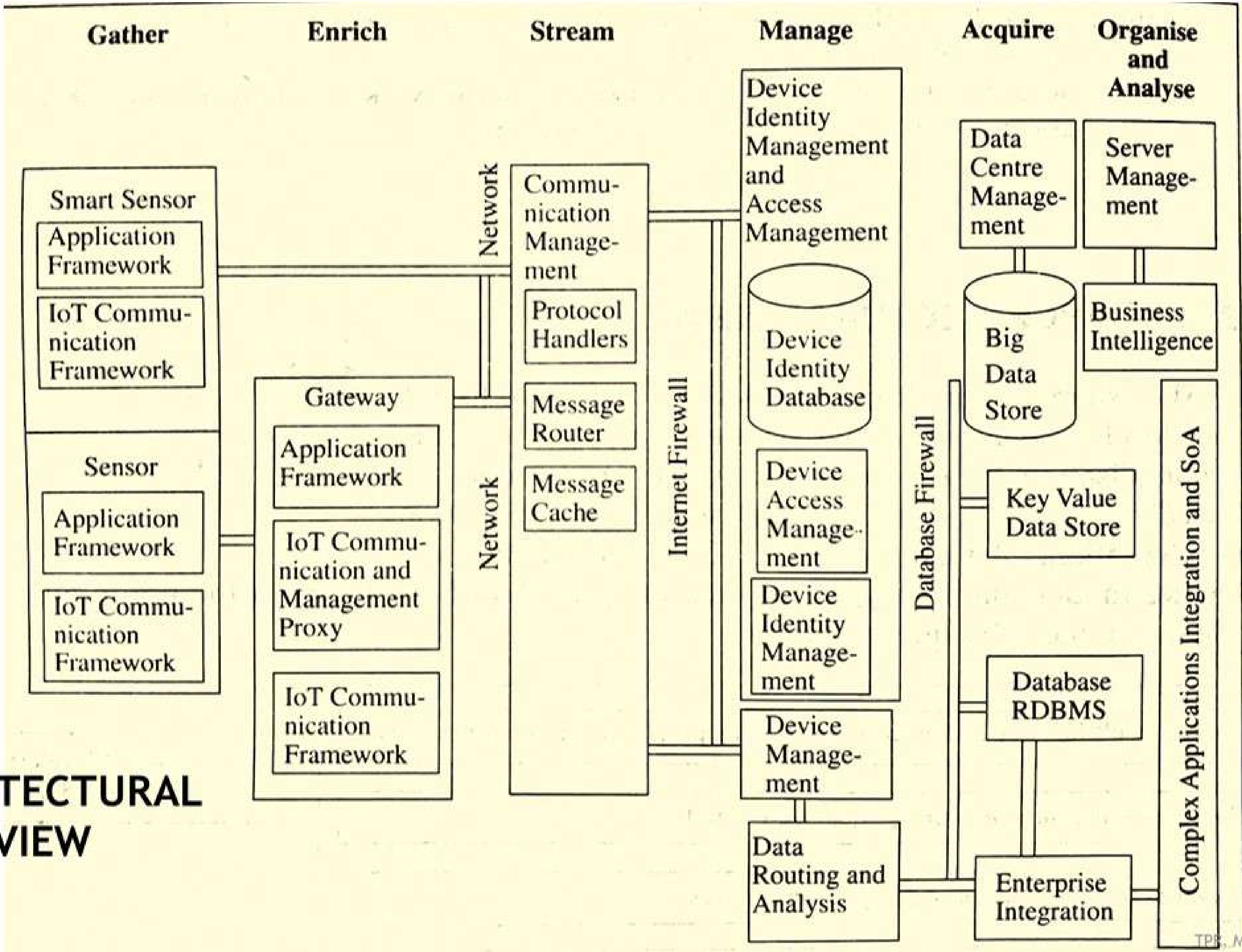
MQTT Protocol

- Message Queuing Telemetry Transport - Open source protocol for M2M/IoT connectivity
- Telemetry - 'Measuring and sending messages to far off places'
- Created by IBM and donated to 'Paho' project of Eclipse
- MQTT Broker - functions as a server node capable of storing messages from publishers & forwards them to clients when needed.

XMPP-Extensible Messaging and Presence Protocol

- **XML** - an open source IETF recommended language - for encoding messages and texts
- XMPP is an XML based specification - also open source

Web Connectivity for connected devices using Gateways



ARCHITECTURAL VIEW

TPR, MACE

Communication Gateway – for connecting web objects

- They connect two application layers, one at sender and other at receiver.
- Gateway also enables use of two different protocols. Eg: IoT device n/w may be Zigbee. The network may connect to web server through a gateway. Eg: Zigbee to SOAP and IP, CoaP protocol conversion gateway for RESTful HTTP
- It facilitates communication b/w web server using TCP/IP protocol conversion gateway and IoT devices
- It also facilitates communication between devices using CoAP client and server using HTTP
- Functions of gateways:
  - Connects sender and receiver using two different protocols. Eg: IoT device n/w maybe using Zigbee for connecting the devices. The n/w then connects to web server using a Gateway. The server interacts and gets data using HTTP
  - Functions as proxy b/w server and system

HTTP

- Application layer in TCP/IP uses HTTP.
- HTTP clients connect to HTTP objects using TCP
- HTTP is a one-way communication - ie from API to server or from server to API
- Polling - method for checking if there are any new messages available from an HTTP server.

SOAP

- Simple Object Access Protocol - protocol for exchanging objects between applications using XML.

REST/RESTFUL applications

- REST architecture constraints and practices are followed by the world wide web now.
- REST provides constraints / rules to data elements, connectors, objects.
- Separation of concern is a feature of REST - ie server is not concerned about client and vice versa - this simplifies the whole implementation, improves scalability
- RESTFUL HTTP API's - use command verbs like GET, POST, PUT, DELETE for communication

Internet Connectivity Principles - Internet connectivity protocols – IP, IPv6, RPL, 6LoWPAN, TCP/IP suite, TCP and UDP

Internet Connectivity

- It is through a set of routers in a global network of routers which carry data a packets as per IP protocol from a source end to another and vice versa

Actions during layer to layer data transmission

- Each layer's processing is as per communication protocol of that layer
- Each layer sends the data from previous layer + a new header thus creating a fresh stack of data for the next layer.
- i.e. Each layer receives data from the previous layer. After required actions, it subtracts the header words and creates a new stack header meant for the next layer
- The process continues till data reaches the last layer
- **Header - words which are required for processing received data at a layer. It has header fields. Each word is 32 bits**

Referring to modified OSI model of IoT, highest layer is application layer & lowest is physical layer

Internet layer uses the IP protocol - IPv4 / Ipv6 / RPL (Routing Protocol)

Internet Protocol (IP)

- that part which contains the address to which, data is to be sent. An **IP address** is a unique **address** that identifies a device on the internet or a local network. In essence, **IP addresses** are the identifier that allows information to be sent between devices on a network.

Transmission Control Protocol (TCP)

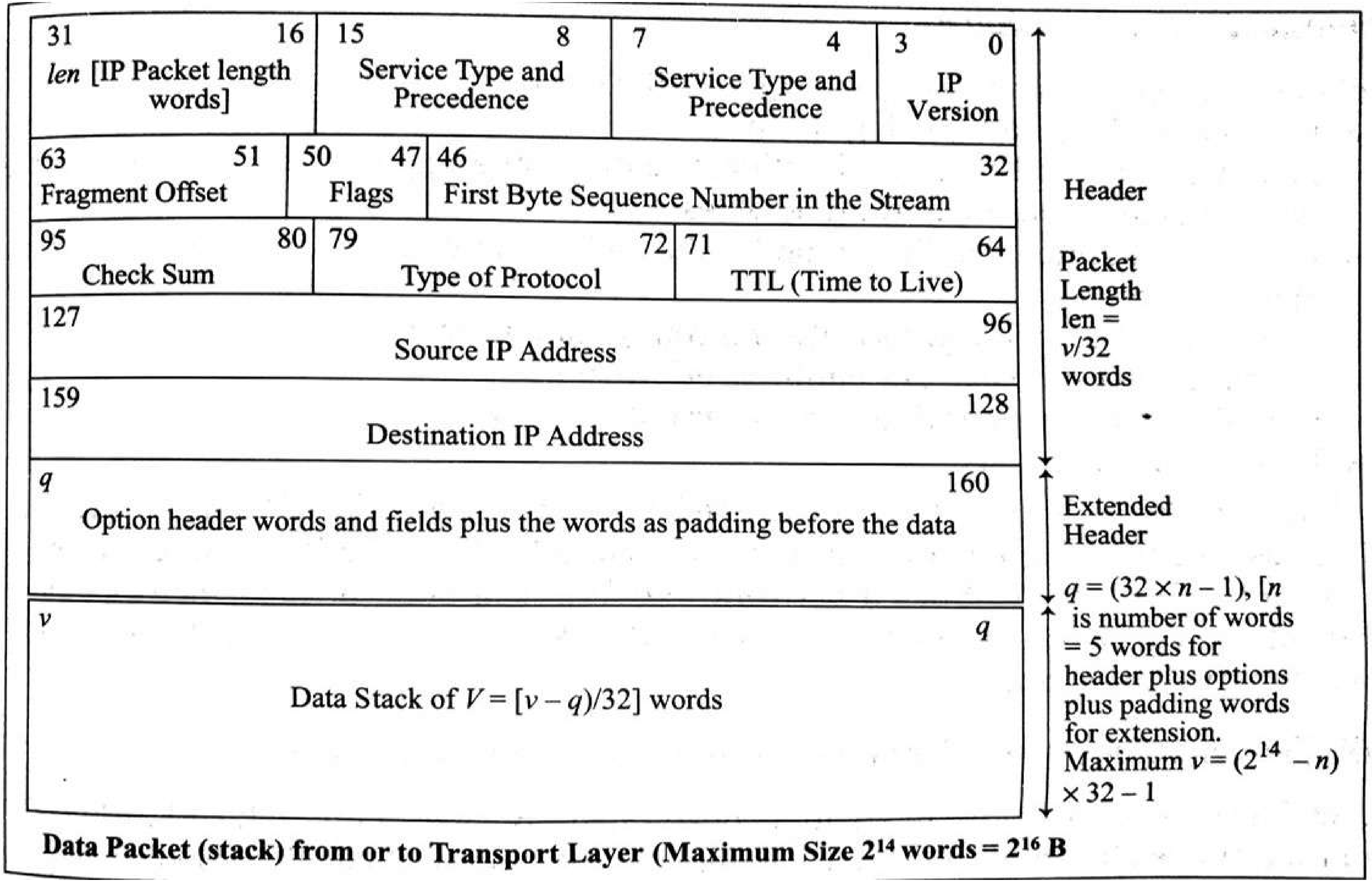
- **TCP** is responsible for data delivery once that **IP** address has been found. **TCP** is **used** in conjunction with **IP** in order to maintain a connection between the sender and the target and to ensure packet order.

Packet routing

- Each router has info on the path to destination. If a number of paths are available, then a number of packets from the same source simultaneously follows different paths. Destination end transport layer reassembles the packet according to their sequence.

Internet Protocol version 4 (IPv4)

- IP – Internet Layer Protocol – refers to the process when a packet of data is transmitted.
- Protocol Data Unit (PDU) is the maximum data unit which can be transmitted/received at layers .1 PDU_{IP} = 1 Packet = 2^{16} Bytes
- Data packet (stack) has size of 2^{14} words = 2^{16} Bytes
- 8 Bits = 1 Byte ; 4 Bytes = 1 word = 32 bits = 4 octets
- IPv4 Data stack has header consisting of 5 words.
  (ie from bits 0 to 159. It can extend if needed.
  Header 4th word = Source IP
  Header 5th word = Destination IP
- Data transmission is unacknowledged
- Each IP layer data stack is called IP packet.
  This packet is guaranteed to reach the destination in TCP, but not guaranteed in UDP
- Address size is 32 bits
- 12 header fields.



31 len [IP Packet length words] 16

15 Service Type and Precedence 8

7 Service Type and Precedence 4

3 IP Version 0

63 Fragment Offset 51

50 Flags 47

46 First Byte Sequence Number in the Stream 32

95 Check Sum 80

79 Type of Protocol 72

71 TTL (Time to Live) 64

127 Source IP Address 96

159 Destination IP Address 128

q Option header words and fields plus the words as padding before the data 160

v Data Stack of $V = [v - q)/32]$ words q

Header

Packet Length len = $v/32$ words

Extended Header

$q = (32 \times n - 1)$, [n is number of words = 5 words for header plus options plus padding words for extension. Maximum $v = (2^{14} - n) \times 32 - 1$

Data Packet (stack) from or to Transport Layer (Maximum Size 2^{14} words = 2^{16} B

IPv6

- Larger addressing space
  IPv6 address has size of 128 bits. Hence, vastly enlarged space
- Hierarchical address allocation
- IPv6 is an alphanumeric addressing method (uses hexadecimal) (IPv4 is a numeric addressing method)
- IPv4 binary bits are separated by a dot(.) whereas IPv6 binary bits are separated by a colon(:)
- IPv4 offers 12 header fields whereas IPv6 offers 8 header fields.
- Manages device mobility, security etc…
- Enables simpler multicast addressing (transmission of data from a single source to multiple recipients)
- Provisions jumbograms (big size datagram, packet containing a payload larger than 65,535 bytes)

RPL – IPv6 routing protocol for Low power Lossy N/w (LLN)

- LPLN refer to constrained nodes network which has low data rate compared to IP
- IETF has given specification for RPL for ROLL network
- IoT/M2M low power lossy environment uses RPL protocol
- Such nodes need to confine communication to the nearest level upwards/downwards

IP Addressing in IoT

- Internet generally uses IPv4.
- IoT/M2M uses IPv6

IP Address

- IPv4 address consists of 32 bits – or – four decimal numbers. Eg:172.18.0.1 - 10101100 00010010 00000000 00000001
- IP addressing can be between 0.0.0.0 to 255.255.255.255
- Total 2^{32} addresses possible
- IP address serves the purpose of uniquely identifying an individual network interface of a host

Static IP

- IP address assigned by Internet service provider

Dynamic IP

- Once a device connects to internet through a router, the router and device use DHCP (Dynamic Host Control Protocol) which assigns an instantaneous IP address to the device.
- When this device disconnects/switches or when router boots again, the dynamic IP is refreshed with a new one.

DNS

- Domain Names System
- Its an application which provides an IP address for a corresponding service from a domain.
- Eg: 198.136.56.2 (32 bit address). Its difficult to use and remember it. Hence, domain names are assigned for such addresses (www.mace.ac.in)
- .com, .org, .in etc… are called top level domain (TLD)

DHCP

- dynamic host configuration protocol
- The DHCP server automatically and dynamically provides IP addresses to each new connected device.

IPv6

- Uses 128 bits
- 32 hexadecimal digits
- 40a0:0acb:8a00:b372:0000:0000:0000:0000
- Unicast – for a single network interface
- Anycast – address of a group of nodes/interfaces – a packet delivered to Anycast address is delivered to just one of the members
- Multicast – address used by multiple hosts – a packet delivered will be delivered to multiple interfaces

Data Acquiring & Storage

TPR@MACE

Data Generation:

- The various methods y which, Data is generated in an IoT environment

1.Passive Devices data

- Data generates at the device/system. A passive device doesn't have a power source. An external power source helps such devices to process and send data. Eg: RFID, ATM debt card

2. Active Devices Data

- Data generates at an active device. These devices will have its own power sources. Eg: streetlight sensor, wireless temperature sensor etcc.. They mostly have a microcontroller, memory & transceiver associated.

3. Event Data

- An event triggers the data. Eg: Traffic congestion data. Water level indicator etc…

4. Device real time data

- Eg: ATM machine – ie the device generates data and communicates in real time

5. Event Driven Device Data

- An event triggers the device data. Eg: When an application seeks information about status of a device, communication happens

Acquiring and storing the data…

TPR@MACE

1. Data Acquisition

- Acquiring data from IoT/ M2M devices.
- There will be an application responsible – interacts and communicates with a number of devices for the acquisition.
- The data will be communicated over the network, transport & security layers
- The application can properly configure the devices for data communication. Eg: To send data at periodic intervals like in a weather station.
- The application can also determine whether to filter and enrich the data or not – through Gateway control. (Gateway is between devices and application).
- Also, transcoding, data management, device management, Privacy, security etc...

2. Data Validation

- Data obtained from devices may not be correct/meaningful/consistent always (Consistency - expected range, pattern & no corruption)
- Validation checks – performed by data validation softwares
- Many techniques can be used – involving logic rules and semantic annotations.
- Appropriate strategies can be used to improve correctness of data – like filtering @ gateways, controlling the rate of acquisition etc…

2. Data categorization for storage

- Data is generally categorized into three types for storage:
  - > Data alone
  - > Data + result of processing
  - > Only results of processing
- Three cases for storage:
  - > Data which require frequent processing in future & therefore only data needs to be stored
  - > Data which needs to be processed only once, but results are used at a later stage – both data and results are stored
  - > Online, real-time or streaming data needs to be processed – only results need be stored
- BIG DATA – data from a large no of devices & sources – stored in databases / cloud based servers. Eg: Smart metering, health services, agriculture etc…

3. Assembly Software for the events

- Event – Any occurrence or situation which is of importance to the process
- A device can generate events. Eg: A pressure sensor in a boiler can generate an event when pressure exceeds the critical value.
- An event can be assigned an ID. On occurrence of an event, a logic value sets/resets. Example
  - >If pressure exceeds, then event has occurred > Logic 1 is generated.
  - >If pressure normal, then event hasn't occurred > Logic 0 is generated.
- Software collects the Logic value + Event ID + Device ID & also adds a date stamp

Data Store – A repository of data objects which integrates into a store

- Data store can refer to a broad class of storage systems including Paper files, Simple files like spreadsheet File systems, Email storage systems (both server and client systems), Databases etc…
- Objects in a data store are modeled using Classes, as per data schemes
- Data store is a general concept - It includes database, relational database, mail servers, directory services etc…
- Data store may be distributed over multiple nodes (ie multiple copies may be stored at different locations to ensure reliability).
  Eg: Apache Cassandra - a free and open-source, distributed database management system designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure.
- Data store is a repository – meaning a group which can be related upon to look for required things
- Database – a repository of data which can be replied upon for reporting, analytics, processing, knowledge etc…

Data Centre Management

- A facility which has multiple banks of computers, servers, large memory systems, high speed internet connectivity
- Provides data security, protection, full data backup with data recovery, power supply backup etc…
- Large industrial units, banks, railways, airlines etc… use data centres.
- Data centres will be dust free, humidification& de-humidification equipment, HVAC equipped, with high level of security.

netmagic NTT

CtrlS Asia's Largest Tier 4 Datacenter

sify

TATA

TATA COMMUNICATIONS

RELIANCE

TULIP®
making it possible

BSNL

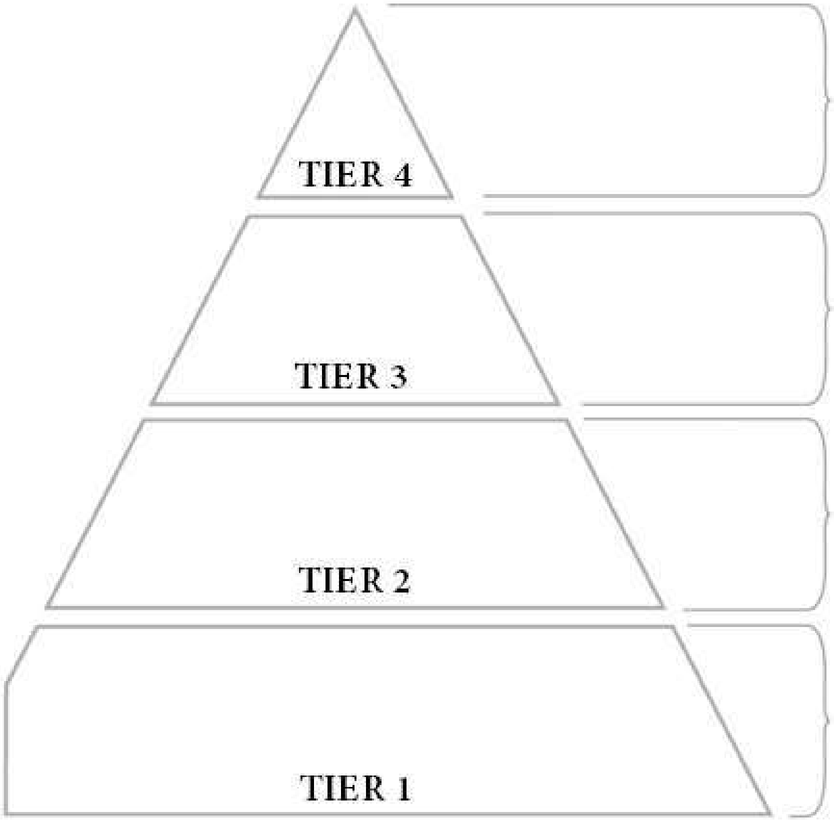
SIS

GPX
Global Systems, Inc.

net4

Web Werks®
DATA CENTERS

Data Center Tiers



Tier 1

- 99.671% Uptime
- no redundancy
- 28.8 Hours of downtime per year.

Tier 2

- 99.749% Uptime
- Partial redundancy in power and cooling
- Experience 22 hours of downtime per year

Tier 3

- 99.982% uptime (Tier 3 uptime)
- No more than 1.6 hours of downtime per year
- N+1 fault tolerant providing at least 72-hour power outage protection

Tier 4

- 99.995% uptime per year (Tier 4 uptime)
- 2N+1 fully redundant infrastructure (the main difference between tier 3 and tier 4 data centers)
- 96-hour power outage protection
- 26.3 minutes of annual downtime.

Server Management

- Managing services, setup & maintenance of systems – round the clock
  - Quick reaction time
  - High security standards
  - Periodic system updates
  - Optimum performance
  - Monitoring of critical services
  - Maintaining confidentiality – protection of customer data from hackers & viruses
  - Strict documentation

Spatial Storage

- Example: when goods are being transported, an RFID tag will be pasted on it. Whenever it goes though a scanner, the location/spatial information is also uploaded into server
- Example: Digital mapping of parking slots in a city – requires information about location and spatial details
- Other examples – internet communication by RFID, ATM, vehicles, ambulances, traffic lights etc…

Organizing the Data

Data can be organized in a number of ways:

- Objects
- Files
- Data Store
- Database
- Relational Database
- Object oriented database

Databases

- Database: a collection of data – organized into tables.
  - Tables provide a systematic way for access, management and update
  - A single table file – flat file – each record is listed separate – unrelated to each other

Relational Database

- Collection of data into multiple tables which relate to each other through special fields(keys)
- They provide flexibility
- Eg: MySQL, PostGreSQP, Oracle database

Object Oriented Database (OODB)

- Collection of objects which save the objects in object oriented design.

Database Management System

- DBMS – software system which contains a set of programs specially designed for creation and management of data stored in a database

In computer science, **ACID** (atomicity, consistency, isolation, durability) is a set of properties of database transactions intended to guarantee validity even in the event of errors, power failures, etc.

In the context of databases, a sequence of database operations that satisfies the ACID is called a transaction.

Atomicity

- An atomic transaction is an indivisible and irreducible series of database operations such that either all occur, or nothing occurs.
- Means a transaction must complete in full, treating it as indivisible. When a service request completes, the pending request filed must be made zero
- Eg: Fund transfer from bank account A to account B. It consists of two operations, withdrawing the money from account A and saving it to account B. Performing these operations in an atomic transaction ensures that money is neither lost nor created if either of those two operations fail.

Consistency

- **Consistency** refers to the requirement that any given database transaction must change affected data only in allowed ways.
- programming errors cannot result in the violation of any defined database constraints
- total bits must be same b/w sent & received

Isolation

- This determines how transaction integrity is visible to other users and systems.
- Or each transaction happens in a distinct order without any transactions occurring in parallel, which can interrupt the original transaction.
- Example: Any reads or writes performed on the database will not be impacted by other reads and writes of separate transactions occurring on the same database.

Durability

- After completion of a transaction, previous transaction cannot be recalled. It guarantees that transactions that have committed will survive permanently.
- For example, if a flight booking reports that a seat has successfully been booked, then the seat will remain booked even if the system crashes.

Distributed Database

- It's a collection of logically interrelated databases over a computer network
- Distributed DBMS – a software s/m that manages a distributed database
- Each system may access all of the data within all of the databases
- Location independent – possible to move data from one physical location to another without affecting users

Consistency – Availability – Partition Tolerance Theorem - **CAP Theorem / Brewer's theorem – it is impossible for a distributed data store to simultaneously provide more than two out of the following three guarantees**

- 'C' – Every user should receive the most recently updated data, irrespective of which node they connect to, for getting the data
- 'A' – Every request receives a response without guarantee that it contains the most recent version of the information. I.e., the user should get a response even if the transaction was unsuccessful
- 'PT' –Partition Tolerance
  Partition refers to a communication break between nodes within a distributed system. (maybe because of network failure, server crash etc…)
  Partition tolerance would mean that the system should still be able to work even if there is a partition in the system. Meaning if a node fails to communicate, then one of the replicas of the node should be able to retrieve the data required by the user.

Query Processing

Query

- An Application seeking a specific data set from a database. Eg: Query at a relational database of a bank server – may be for an ATM transaction done by a specific user at a specific month

Query Processing

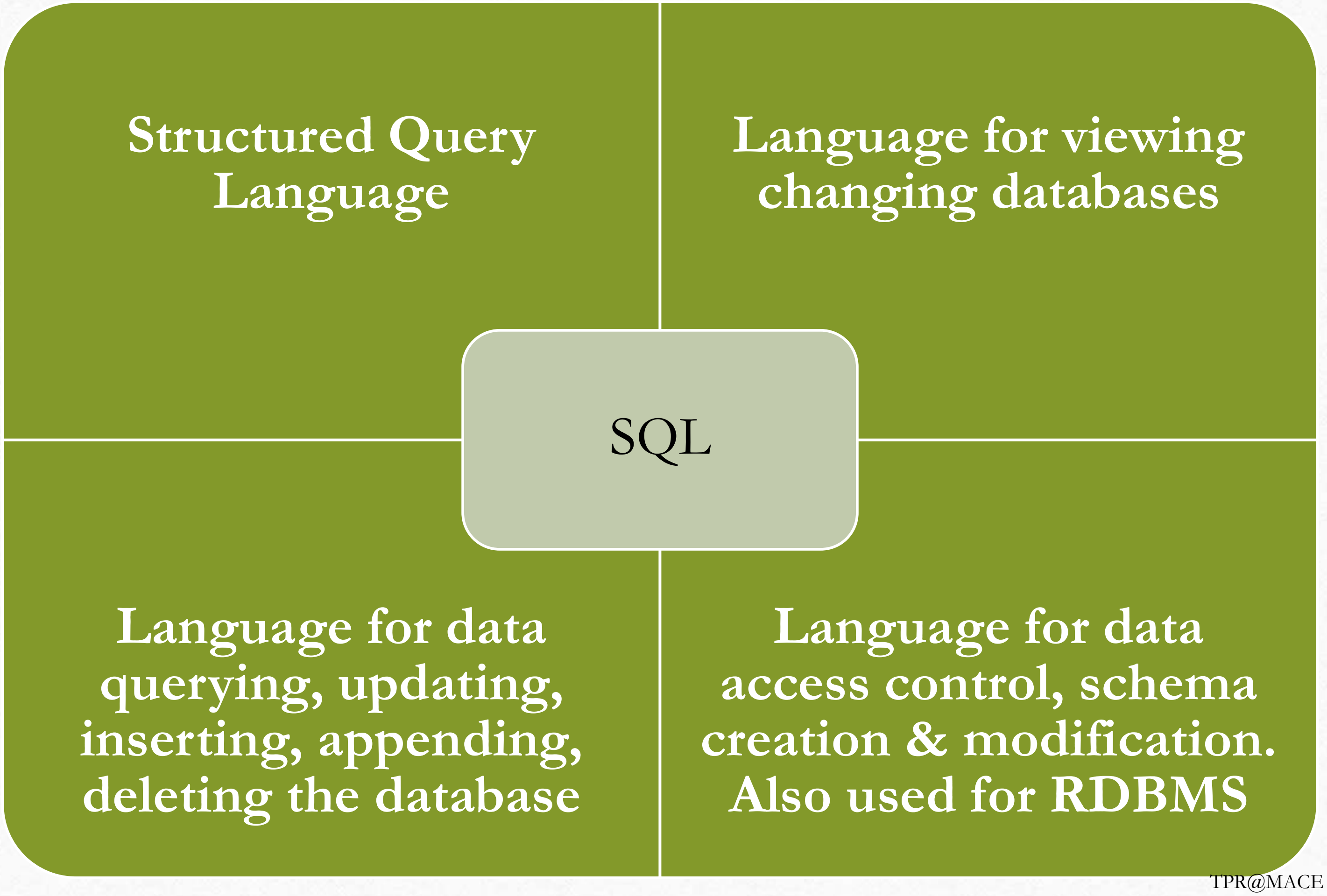
- Using a process and getting the results of the query made from a database.

Distributed Query Processing

- Query processing operations in a distributed database on the same system/networked system

Steps in Query Processing

Parsing & Translation	• Translate the query into an internal form – a relational algebraic expression & then parser – check syntax and verifies relations
Decomposition	• To complete the query process into micro – operations using analysis, normalization & semantic analysis. Check whether that **query** is syntax correct and semantically correct
Optimization	• Optimising the cost of processing – ie the number o f micro-operations generated in processing
Evaluation Plan	• A query execution engine/software – takes a query evaluation plan & executes that plan
Returning	• Returning the results



SQL

Structured Query Language

Language for viewing changing databases

Language for data querying, updating, inserting, appending, deleting the database

Language for data access control, schema creation & modification. Also used for RDBMS

Analytics

- Organized data after acquiring can be used for multiple purposes.
- In general, applications use data for monitoring, reporting and rule-based actions
- Example of analytics in the application & n/w domain of ATM's connected to a bank server …

Analytics Phases – three phases before deriving new facts & providing business intelligence

- **Descriptive analytics** – enables deriving additional data from visualisations & reports
  - Eg: company reports that simply provide a historic review of an organization's operations, sales, financials, customers, and stakeholders
- **Predictive analytics** – enables extraction of new facts - then predicts / forecasts with the help of identifying patterns, clusters, anomaly detection etc…
- **Prescriptive Analytics** – enables derivation of additional value & undertake better decisions to maximize profits. Not only does it address anticipated things, it will analyze why something has happened and take corrective steps
  - Eg: Self-driving automobiles (Google, Tesla)

BIG DATA

An extreme amount of data – high volume, variety, velocity (3V's) & also Veracity (4V's)

Volume – Data absorbed is beyond the capability of commonly used softwares to analyse

Variety – structured & unstructured data in different formats

Velocity – data received at higher rates due to many sources

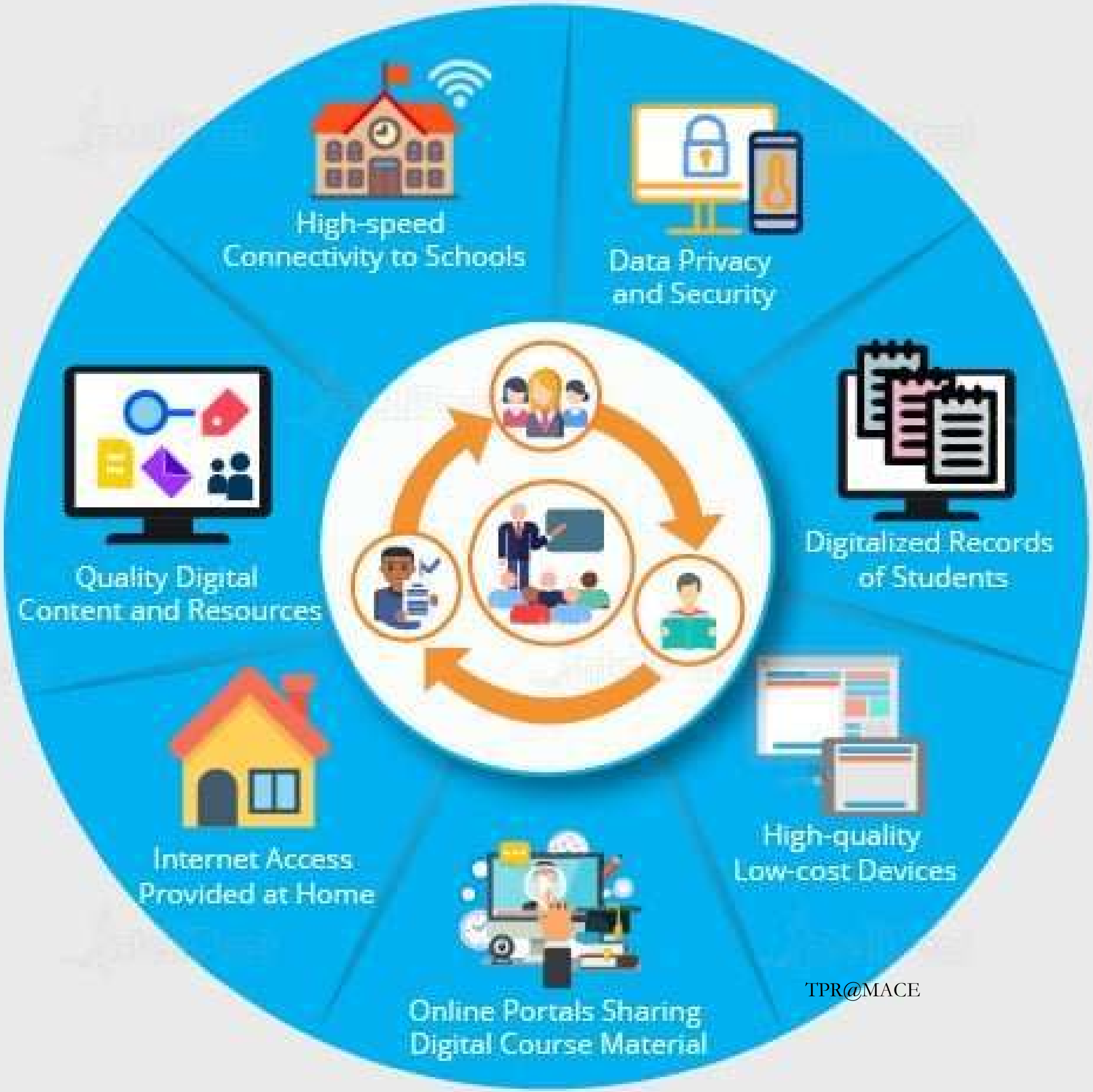
Veracity – variation in data quality for analytics.
ie how accurate and applicable is the data

Big Data Analysis

- Open Source softwares – Hadoop and MapReduce from Apache – enable storage and analyse the massive data
- The following are used for bigdata analytics in Hadoop Ecosystem
  - >Hadoop File Sytem (HDFS)
  - >Mahout –a library of machine learning algorithms
  - >HiveQ – an SQL like scripting software
- It is an open source framework, storing and processing big data
- Clusters of computer nodes process that data using simple programming models.
- Processing takes place in a distributed environment.

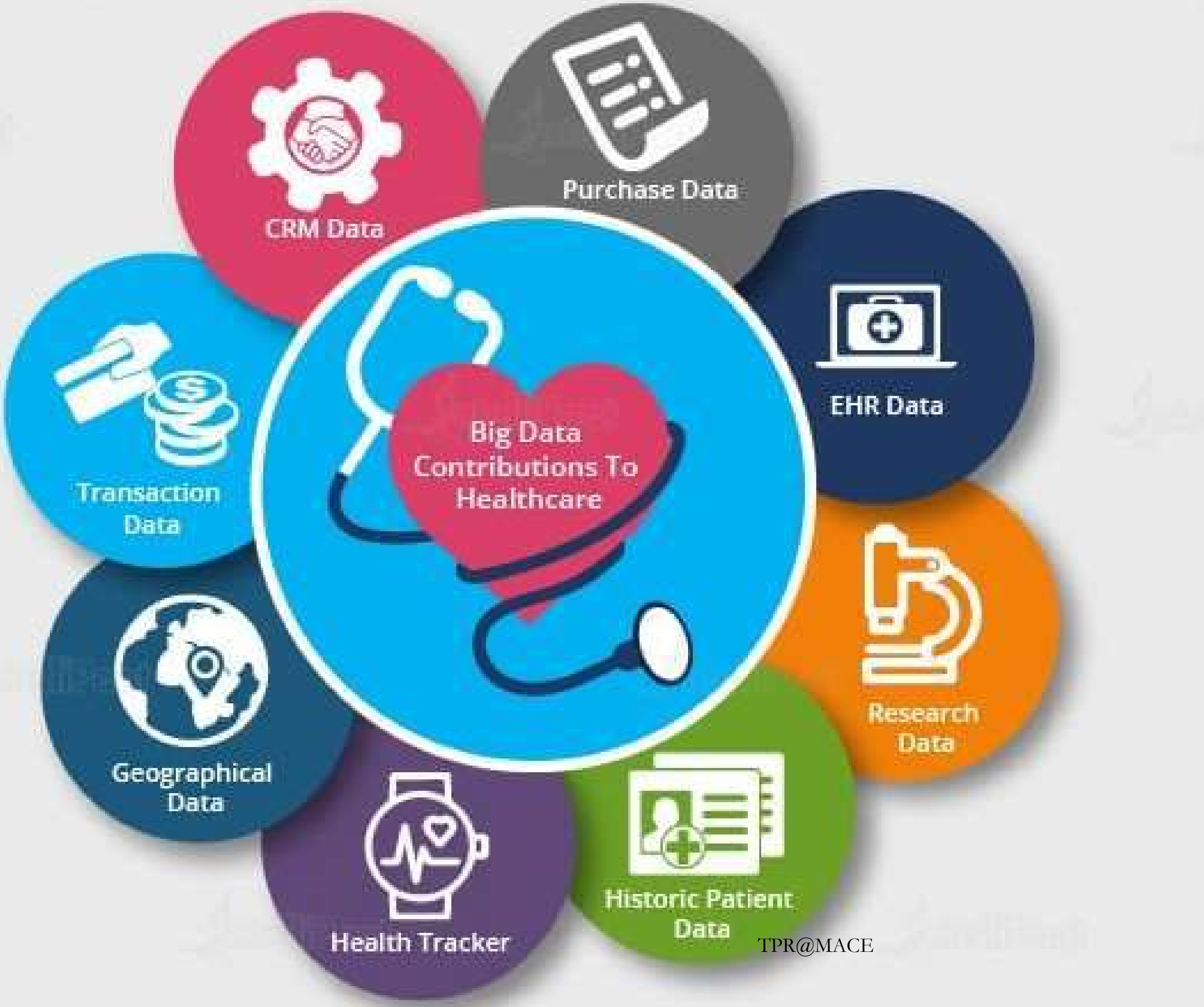
Education

- Customized and Dynamic Learning Programs
- Reframing Course Material
- Grading Systems
- Career Prediction


High-speed Connectivity to Schools
Data Privacy and Security
Digitalized Records of Students
High-quality Low-cost Devices
Online Portals Sharing Digital Course Material
Internet Access Provided at Home
Quality Digital Content and Resources

Healthcare

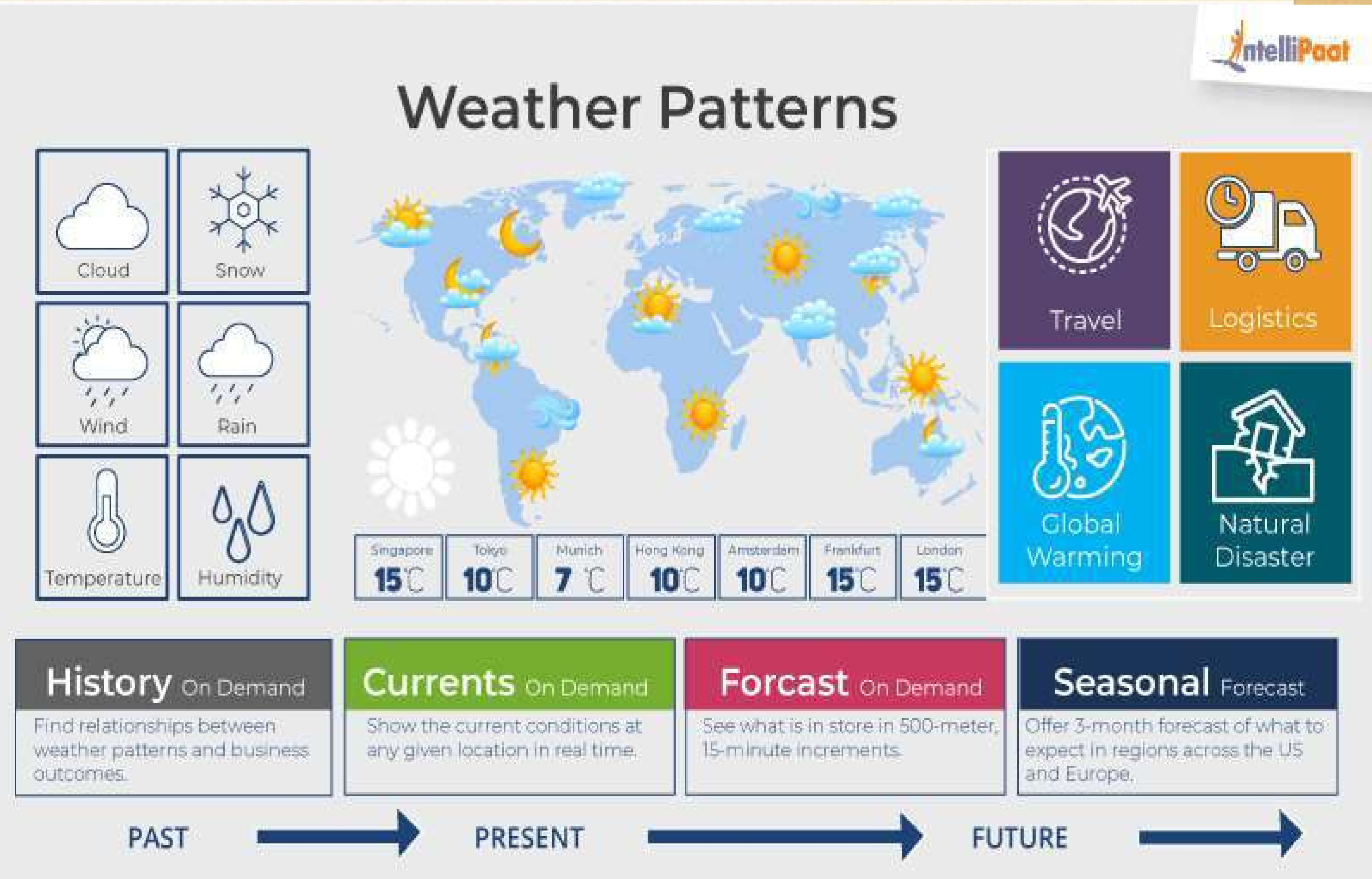
- Better diagnostics
- Prevents outbreaks of epidemics
- Avoid preventable deceases by early detection
- Evidence based medicine based on past medical records



TPR@MACE

Weather

- In weather forecasting
- To study global warming
- In understanding the patterns of natural disasters
- To make necessary preparations in the case of crises
- To predict the availability of usable water around the world



Weather Patterns

History On Demand: Find relationships between weather patterns and business outcomes.

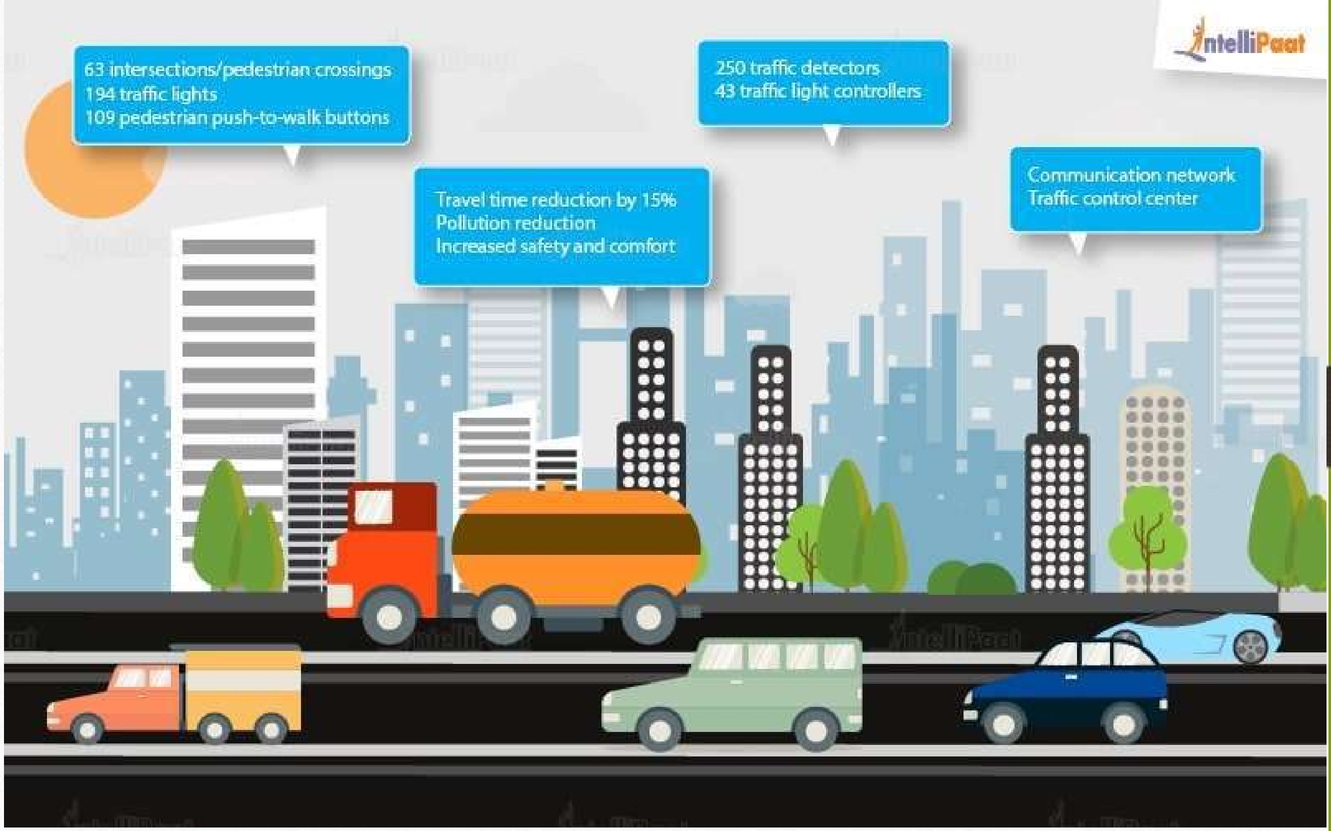
Currents On Demand: Show the current conditions at any given location in real time.

Forcast On Demand: See what is in store in 500-meter, 15-minute increments.

Seasonal Forecast: Offer 3-month forecast of what to expect in regions across the US and Europe.

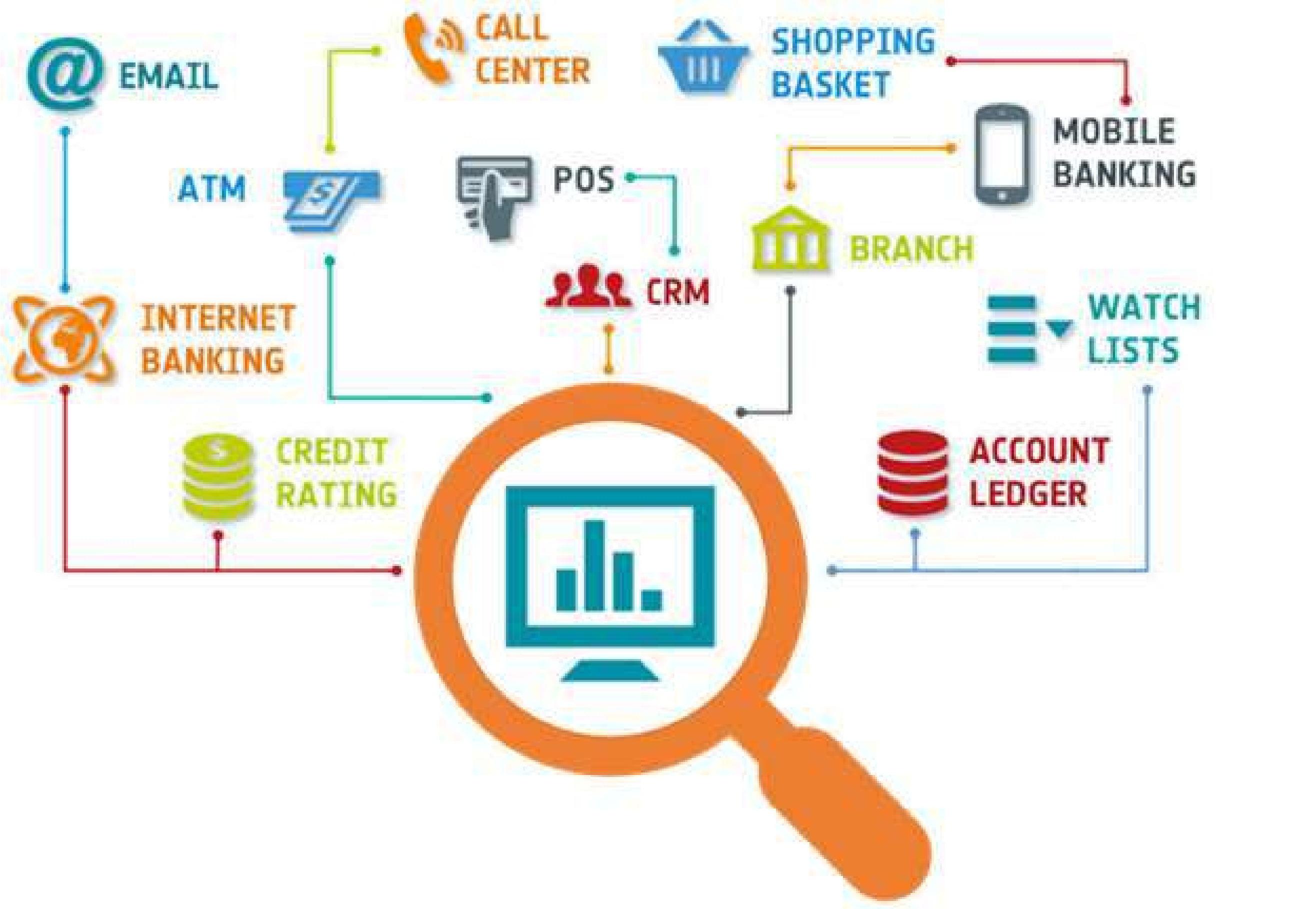
Transportation

- Route planning
- Congestion management and traffic control
- Safety level of traffic


63 intersections/pedestrian crossings
194 traffic lights
109 pedestrian push-to-walk buttons
250 traffic detectors
43 traffic light controllers
Travel time reduction by 15%
Pollution reduction
Increased safety and comfort
Communication network
Traffic control center
IntelliPaat

Banking

- Misuse of credit/debit cards
- Venture credit hazard treatment
- Business clarity
- Customer statistics alteration
- Money laundering
- Risk mitigation

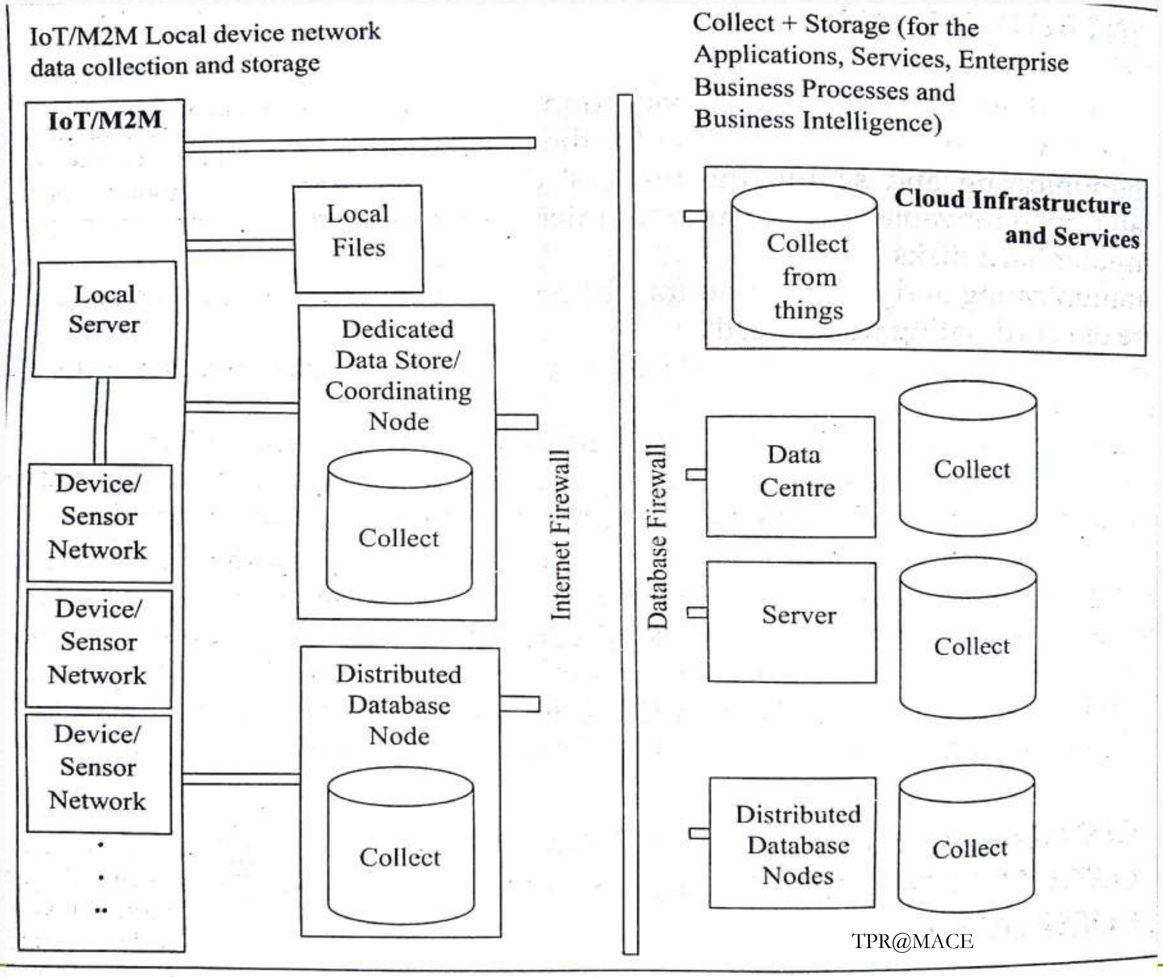


Cloud Computing for data storage

Conventional methods of data collection and storage

- Saving device's data at local server
- Saving in files on locally removable media like SD cards, hard-disks etc…
- Communicating and saving data and results of computations in a dedicated data store & coordinating that node locally
- * At a local node in a distributed data store
- * At a remote node in a distributed store
- Communicating on the internet and saving at a data store in a web/enterprise server

Different methods of data collection, storage and computing


IoT/M2M Local device network data collection and storage
IoT/M2M
Local Server
Device/ Sensor Network
Device/ Sensor Network
Device/ Sensor Network
Local Files
Dedicated Data Store/ Coordinating Node
Collect
Distributed Database Node
Collect
Internet Firewall
Database Firewall
Collect + Storage (for the Applications, Services, Enterprise Business Processes and Business Intelligence)
Cloud Infrastructure and Services
Collect from things
Data Centre
Collect
Server
Collect
Distributed Database Nodes
Collect

Resources

- One that can be read, written or executed. A path specification. Eg: Data point, pointer, data object, etc…

Environment

- A set of conditions / for programming, program execution

Platform

- Basic hardware, OS and N/W which uses the software applications/services

Edge computing

- A type of computing that pushes the frontier of applications, data and services away from centralized nodes to IoT data generating nodes, ie to the extremes of networks
- This enables the usage of resources at edges/extremities, which will be helpful in case of low power lossy networks

CLOUD COMPUTING

- A collection of services available over the internet.
- Cloud delivers the computational functionality
- Like users of electricity don’t know the source or infrastructural details, a user of cloud computing services need not know how the infrastructure deploys or the details of computing environment.
- Like users don’t know what processor is inside a computer, a cloud computing user can use the computing and intelligence in the cloud

Cloud platform offers the following:

- Infrastructure for large data storage of devices, RFID's, industrial & automobile data
- Computing capabilities such as analytics, IDE (Integrated Development Environment)
- Collaborative computing and data sharing

Internet Cloud + Clients = User Application & services with ' no boundaries & walls'

Virtualization

- Virtualized environment provided by cloud storage and computing environment
- Enables applications & services to execute in an independent environment – each storing & executing in isolation on the same platform.
- Applications need not be aware about the platform – just internet connectivity required
- >Such storages – cloud storage
- >Such computing – cloud computing

Virtual storage example – Apple iCloud

Network Function Virtualization – many users access resources appearing as just one network

Virtual Server – user applications access not only one , but many servers

Virtualized desktop – user applications can change / deploy multiple desktops though access by user is through their own OS

Cloud computing and Virtualization

- **Cloud Computing** is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.
- **Virtualization** is a Software that creates "separated" multiple images of the hardware and software on the same machine. This makes possible to install multiple OS, multiple software and multiple applications on the same physical machine.
- Cloud is good for public use. IT companies use Virtualization for cost-efficient data center setup.

Cloud computing features & advantages

- On demand self service to users for storage, servers etc…
- Resource pooling
- Broad n/w accessibility
- Scalability
- Maintainability
- Virtualisation
- Resilient
- Security

Concerns

- Requirement of high speed internet connection
- Limitations on services available
- Possible data loss
- Different API's and protocols used in different clouds
- Loss of user's control

Public cloud

- Provisioned by educational institutions, industries, govt. institutions, business & is open for public use

Private cloud

- Use by private institutions – organisations – employees

Community clouds

- For use by a community formed by institutions, industries, business
- Community will specify the security & compliance considerations

Hybrid Cloud

- A set of two or more distinct clouds (public, private or community) with distinct data stores & application – to deploy proprietary technology

SENSOR TECHNOLOGIES

Data is generated using sensors , embedded devices and systems at the physical layer in the IoT architecture

This data then communicates through the following layers: >data-link > data adaptation > network > application support > application layer

The following will be discussed in this chapter:

- Data Sensing
- Participatory sensing in M2M, Industrial IoT, automotive IoT
- Actuators
- RFID's
- WSN's

Sensor technologies

- Designing sensors, Associated electronics readers, circuits, devices
- Sensors basically senses a change in physical parameters like temperature, pressure, light, smoke, metal etc…
- Also they sense acceleration, orientation, location, vibrations
- They convert physical energy like heat, sound, strain etc… into a corresponding electrical energy signal.

Smart Sensors

- They have electronic circuits within themselves & can perform computing & communication.
- The circuit receives energy in the form of variations in current, voltage, phase angle or frequency
- Analog sensors will have a reference against which this measurement is made

Examples of sensors having Resistive Element

- Resistance of a platinum wire changes with temperature
- Resistive variation of a touch screen
- Strain sensor – Change in R with application of a strain
- Gas sensor – metal oxide coated sensor whose R varies with vapour adsorption
- Photo-conductors – R changes with application of light

Examples of sensors having Capacitive Element

- Used as a proximity sensor when capacitance varies when the sensor comes in contact with a hand/object
- Capacitive tough sensor – variation in capacitance caused by finger position

Transistor based

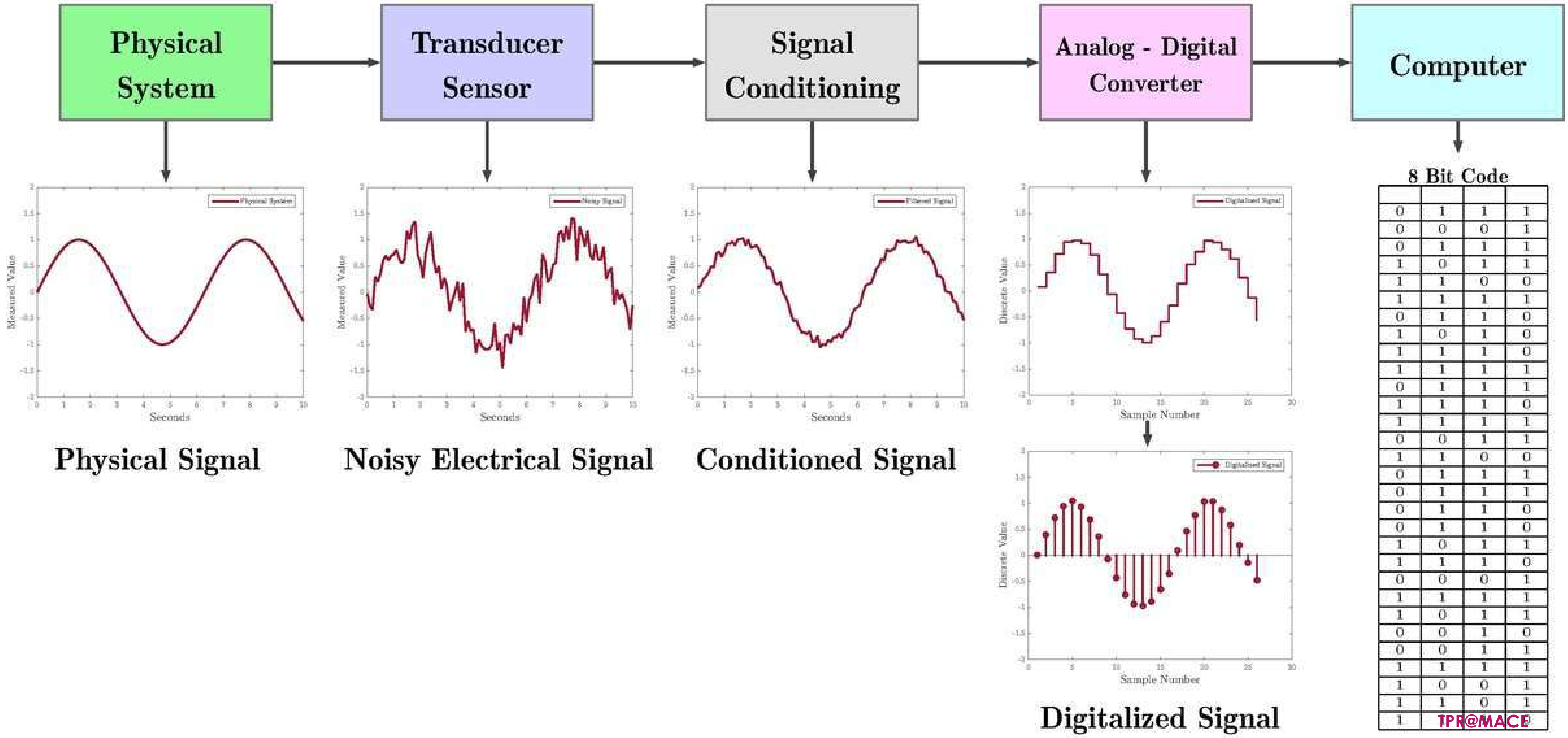
- Phototransistors – BJT's with special windows for light to enter shows variation in I_{sat}
- Photodiode

Sensors in a smart phone ??

Analog Sensors

- Sensor + Electronics
- Eg: Temperature, strain, pressure, force, flex, vapours, mag field, proximity
- Sensor O/P >> I/P of signal conditioner >> ADC >> digital o/p >>read by micro-controller

Digital Data Acquisition System



0	1	1	1
0	0	0	1
0	1	1	1
1	0	1	1
1	1	0	0
1	1	1	1
0	1	1	0
1	0	1	0
1	1	1	0
1	1	1	1
0	1	1	1
1	1	1	0
1	1	1	1
0	0	1	1
1	1	0	0
0	1	1	1
0	1	1	1
0	1	1	0
0	1	1	0
1	0	1	1
1	1	1	0
0	0	0	1
1	1	1	1
1	0	1	1
0	0	1	0
0	0	1	1
1	1	1	1
1	0	0	1
1	1	0	1
1	[illegible]	[illegible]	[illegible]

Temperature measurement using resistance sensor ??

Capacitive sensor ??

Serial Ports

- Serial ports are commonly used for communication b/w microcontroller and ADC

ADC

Microcontroller accepts data from an ADC

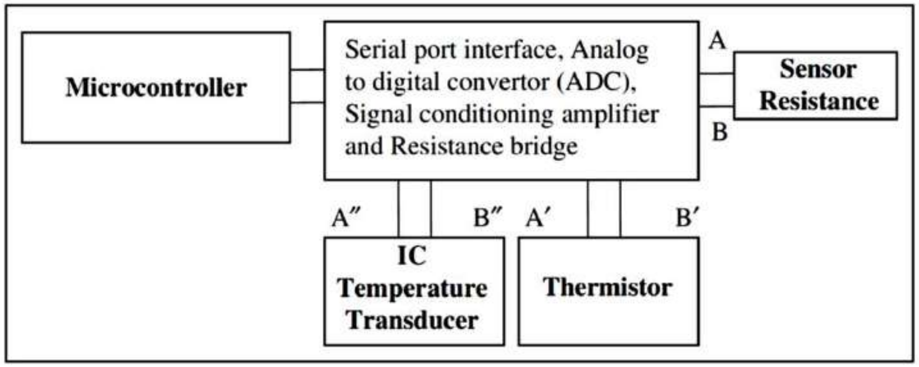
- An 8 bit port accepts 8 bit input = 0 to 255 decimals
- If it's a 12 bit data, it is taken in two batches – 8 + 4 bits = 4096 decimals

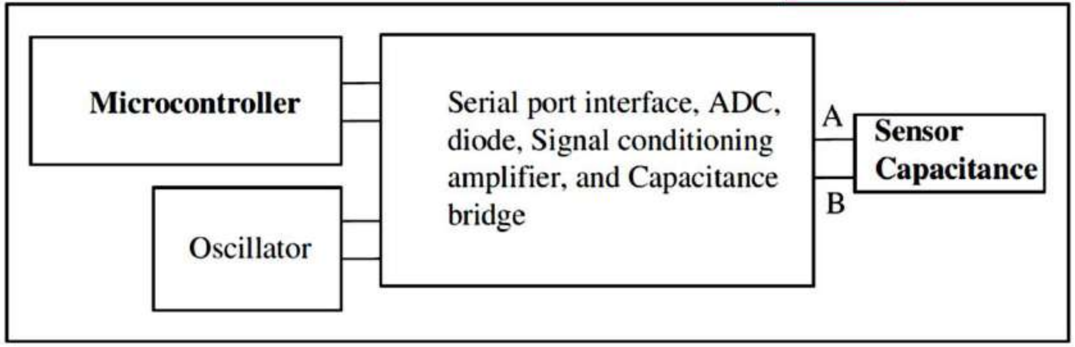
Sampling in ADC

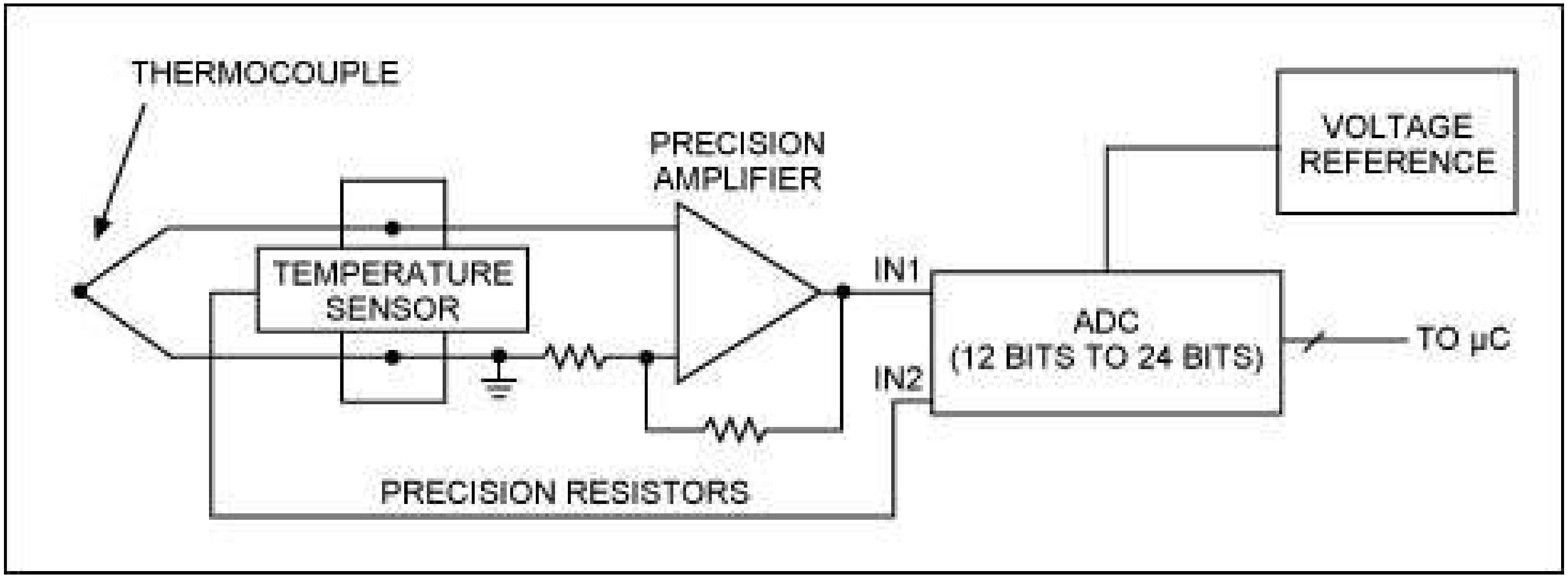
- Rate at which, ADC accepts input
- Eg: Music recording – 16 bit - 48kHz is DVD quality

Signal Conditioning

- Accepts input – adds/subtracts an offset voltage

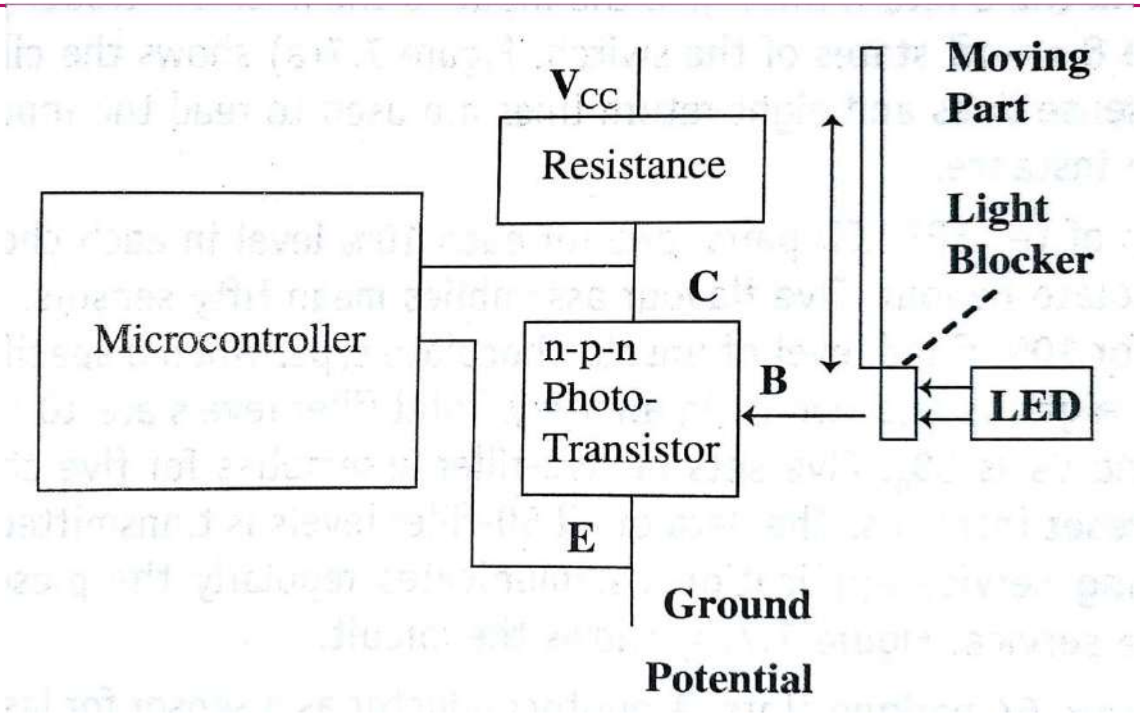

Microcontroller
Serial port interface, Analog to digital convertor (ADC), Signal conditioning amplifier and Resistance bridge
A
B
Sensor Resistance
A″
B″
A′
B′
IC Temperature Transducer
Thermistor


Microcontroller
Oscillator
Serial port interface, ADC, diode, Signal conditioning amplifier, and Capacitance bridge
A
B
Sensor Capacitance


THERMOCOUPLE
TEMPERATURE SENSOR
PRECISION AMPLIFIER
IN1
IN2
VOLTAGE REFERENCE
ADC (12 BITS TO 24 BITS)
TO μC
PRECISION RESISTORS

Digital Sensors

- Gives digital o/p
- 0 or 1 (on-off state)
- Output of 0's and 1's (Binary output - digitial data)
- Can be used for sensing a situation or sudden change

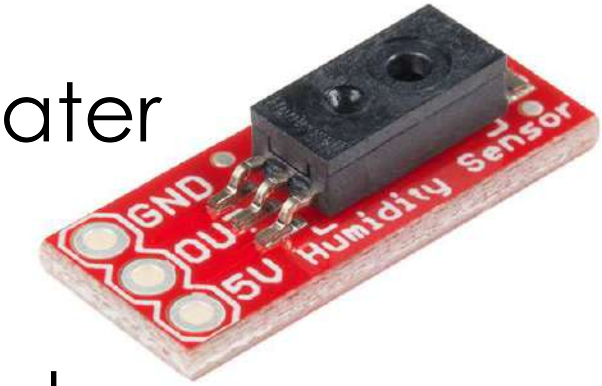


Temperature Sensor

- **Thermistors**
  - Large resistance variations for narrow temperature changes
  - Two types – PTC & NTC
  - PTC – eg: Thin wire of Platinum, metal alloys, doped polycrystalline ceramic. 0-1600 [degrees Celsius range]
  - NTC – eg: ferric oxide (Fe_2O_3) with titanium (Ti) doping
  - Certain IC's – AD590 function as temp transducer & generates micro amps for every 1 degree temp change

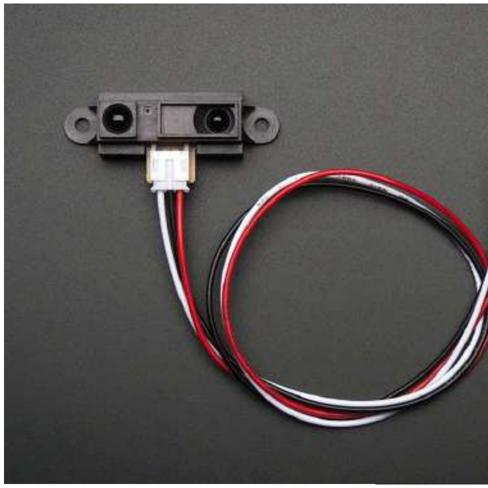
Humidity

- Measured as relative percentage ratio (RH%) of content of water vapour in air compared to the situation of maximum possible water vapour content for the air temperature at the instance.
- Capacitor sensor commonly used - change in capacitance as humidity changes



Distance

- **Infrared Sensors**
  - 0.15m to 0.8 m range
  - Principle – A beam of IR radiation sent by an IR LED – travels, hits an object and is received by a phototransistor. The distance travelled = 2 x actual distance
  - Reflected radiation delay = 2 x 3.3n sec per metre
  - Above 0.8m, the reflected wave intensity insufficient for detection
  - Below 0.15m, the reflected wave reaches much faster than 1n sec – cannot detect.

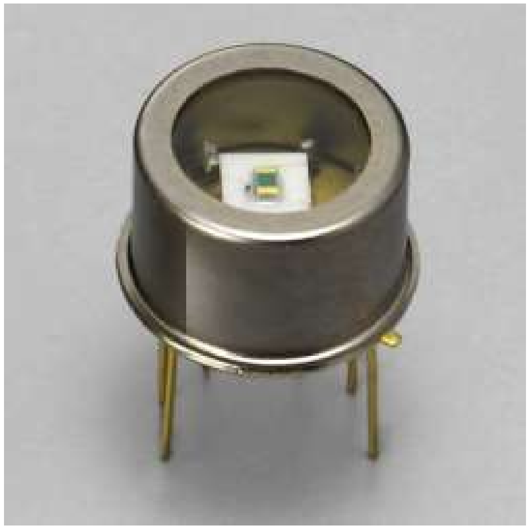


Distance

- **Ultrasonic Sensors**
  - Sends pulses of freq ~ few kHz
  - Detects the echo signal
  - Wave delay = 2 x m milli sec / meter in air
  - Long range measurements possible
  - Industrial automation, rail tracks, oil pipeline faults etc…

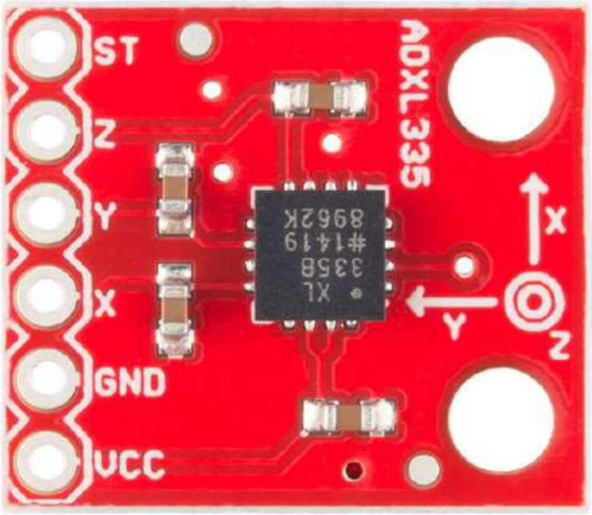
Light

- Photoconductors
  - Sensor shows drop in resistant with surrounding light
  - PN junction diodes with their junction exposed
  - Figure shows Indium Antimonide Sensor

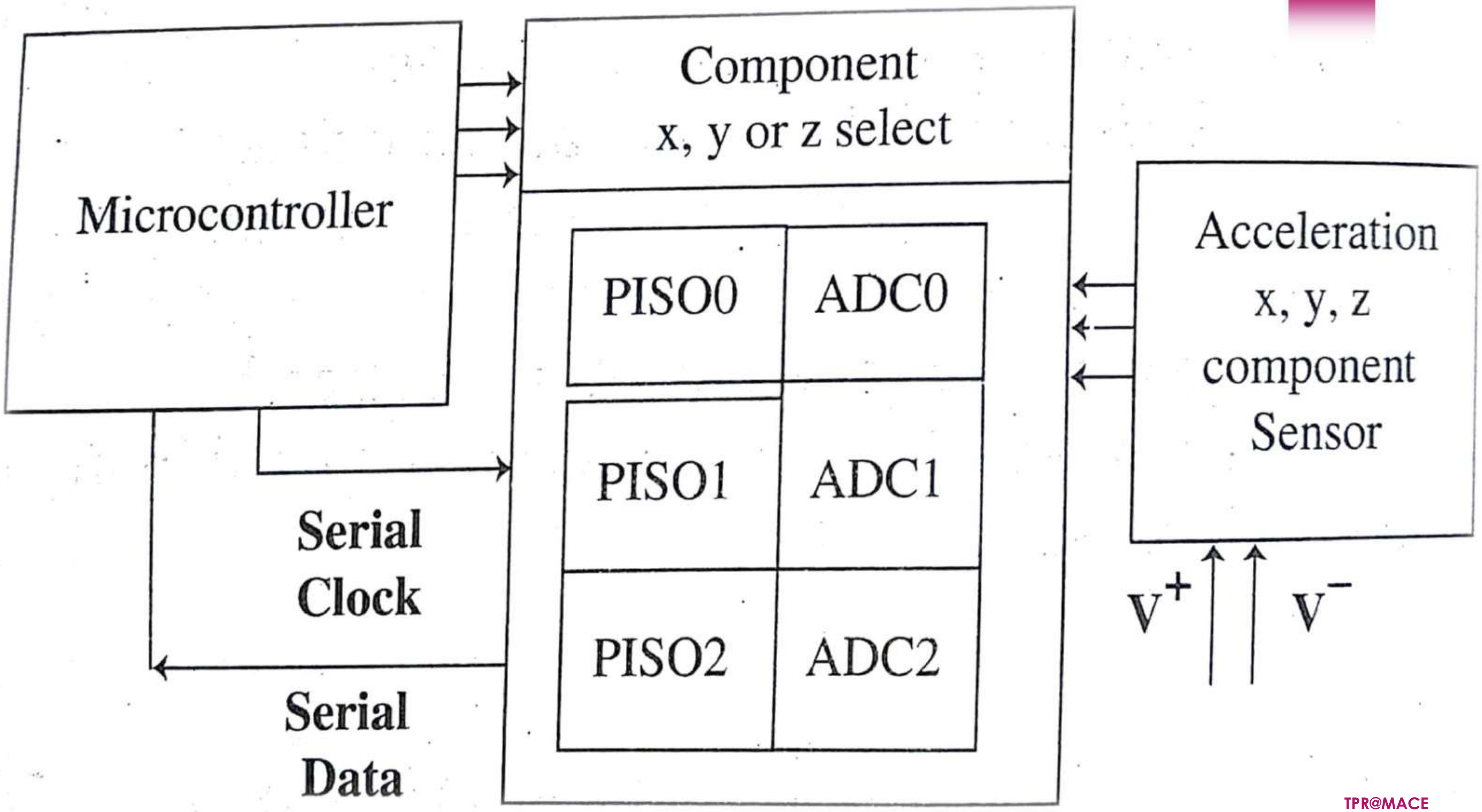


Acceleration

- Micro-Electro-Mechanical Sensors (MEMS) detects acceleration - ax, ay, az
- Corresponding movements in each axes causes variations in three capacitance values Cx, Cy, Cz – since their distance varies
- Capacitances form part of an electronic circuit – the voltages give ax, ay, az

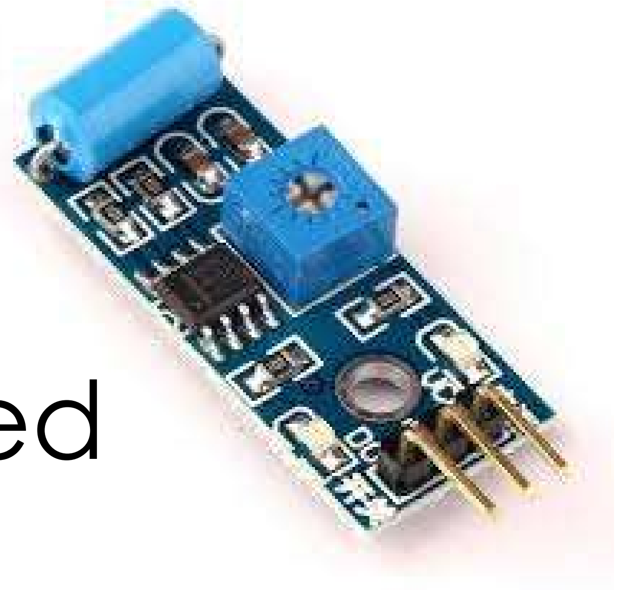


Sparkfun – ADX335


Microcontroller
Component
x, y or z select
PISO0
ADC0
PISO1
ADC1
PISO2
ADC2
Acceleration
x, y, z
component
Sensor
Serial
Clock
Serial
Data
V^+
V^-

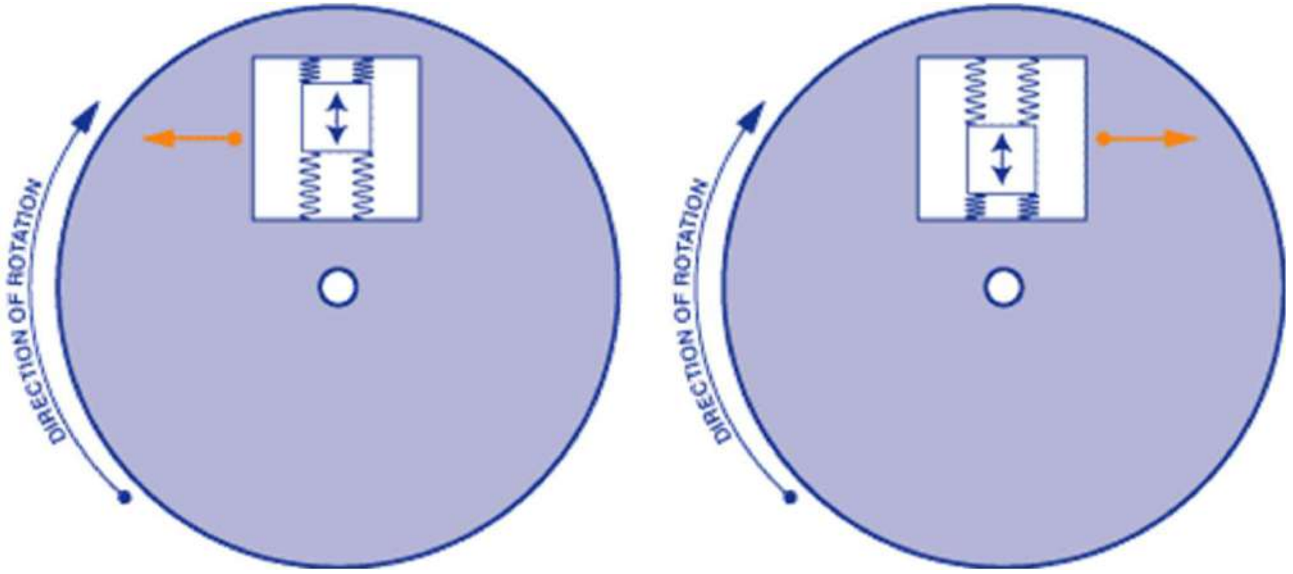
Vibrations and Shock

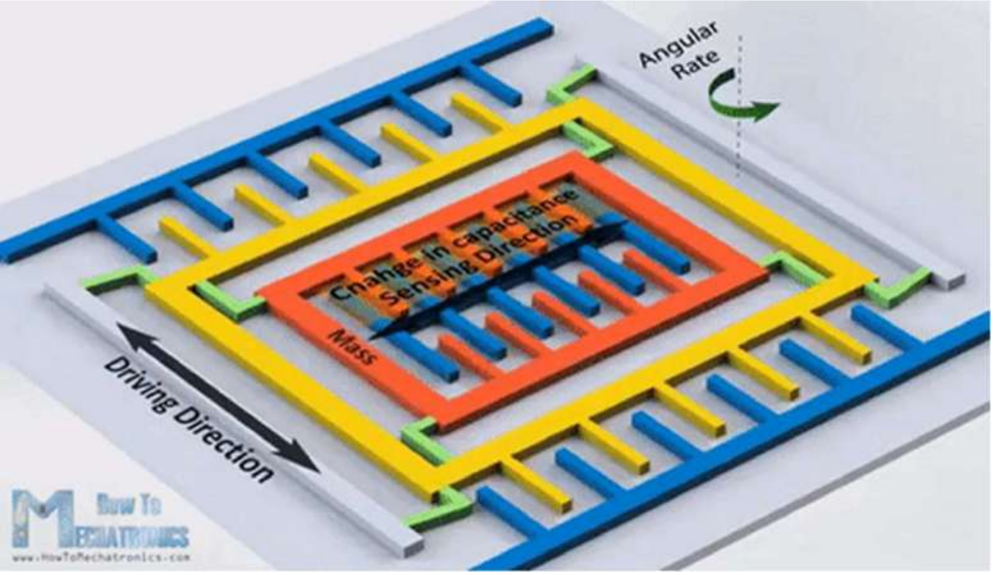
- MEMS using piezoelectric effect instead of capacitance
- Vibrations create repeated compressions / decompressions
- Translate to a voltage

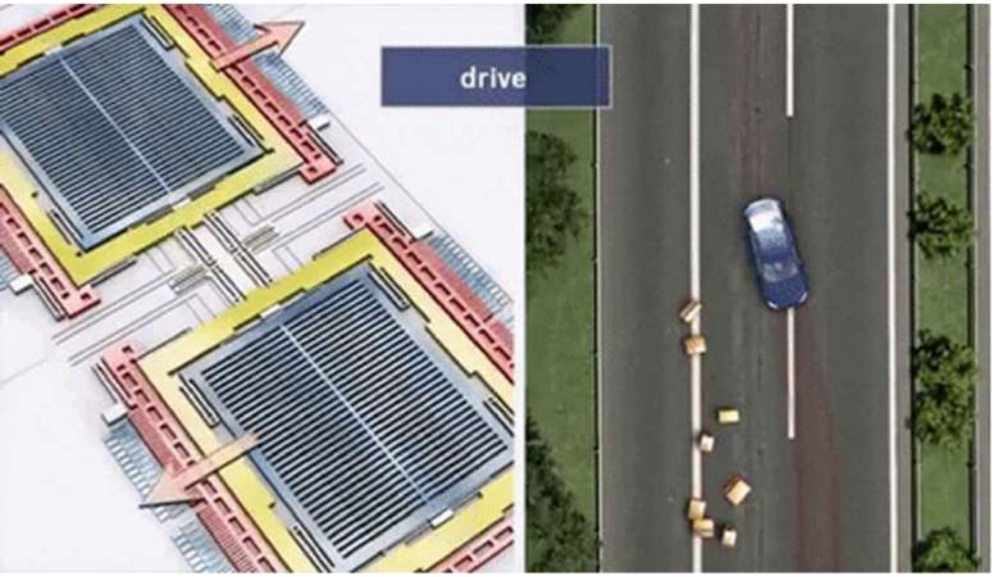


Angular Acceleration Sensor

- Gyroscope
- Measures change in angular velocity & angle
- MEMS gyroscopes




Angular Rate
Cnahge in capacitance
Sensing Direction
Mass
Driving Direction
How To Mechatronics
www.HowToMechatronics.com


drive

Orientation & direction compass

- Digital compass
- Consists of gyroscopes having a magnetic strip which aligns according to external field

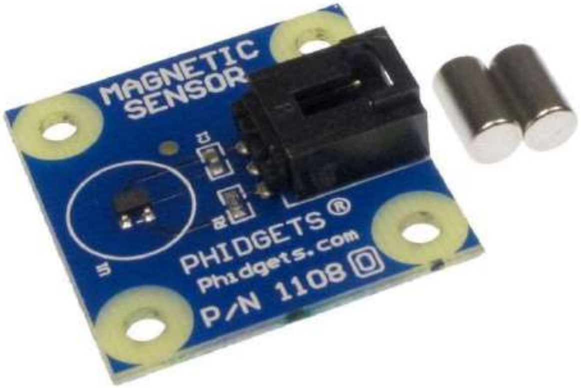
Magnetic sensors

Electric current

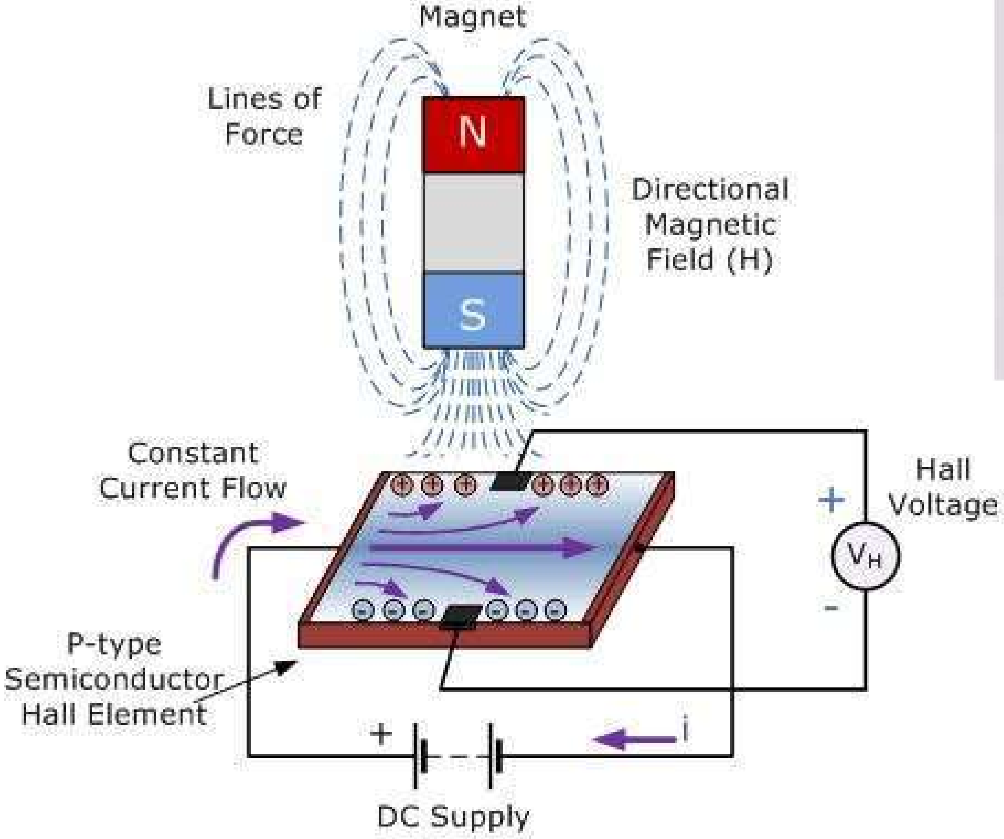
- AC – miniature transformer
- DC – Hall sensors

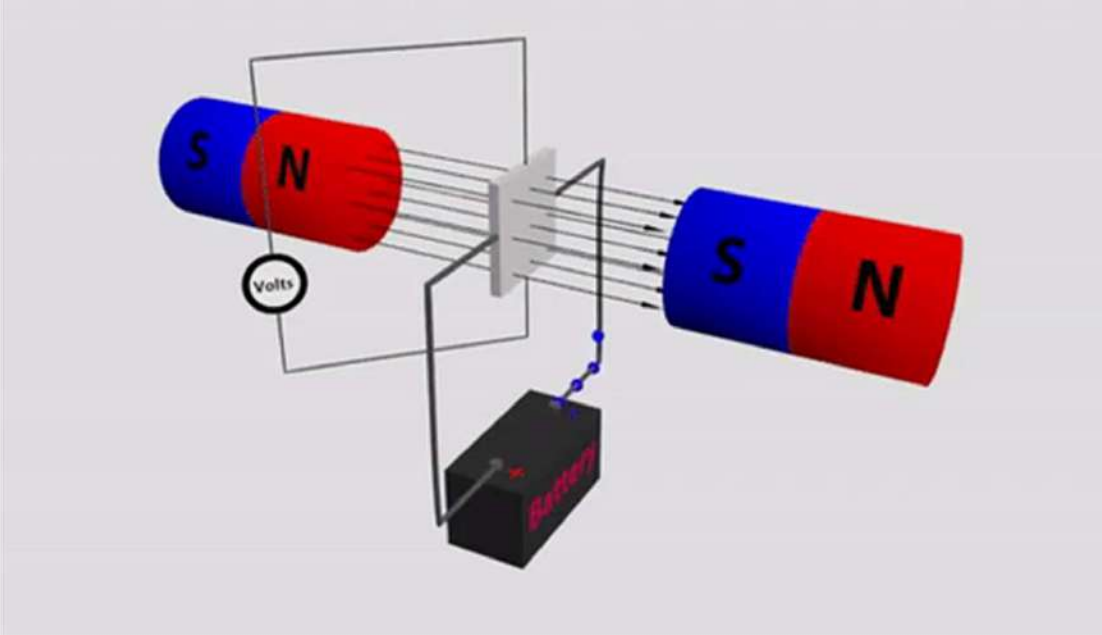
Sound

- Microphone



Current Sensor using Hall effect

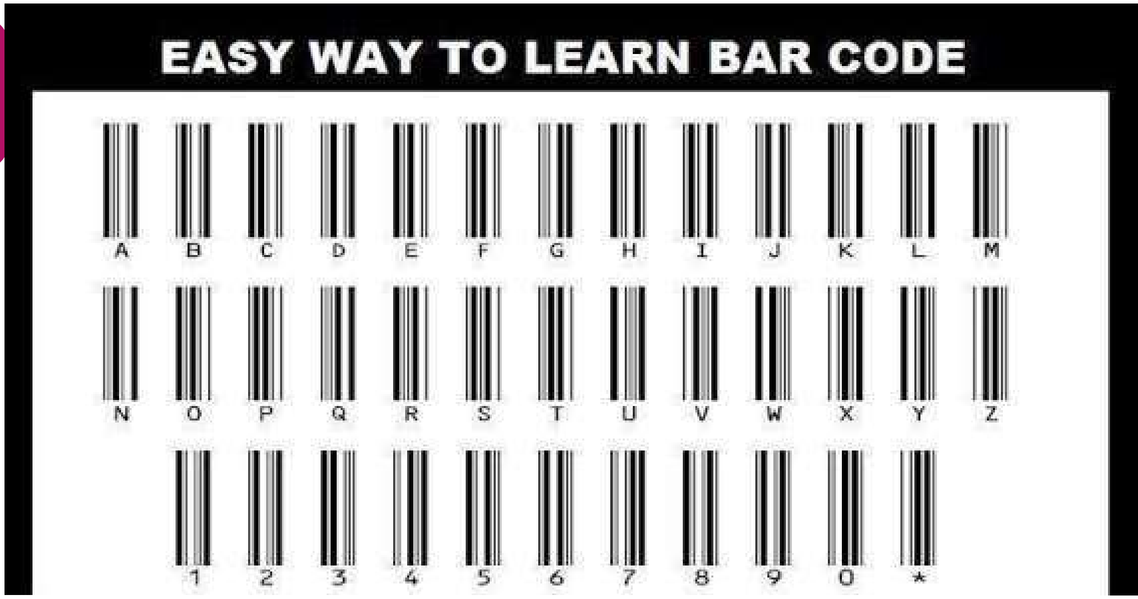


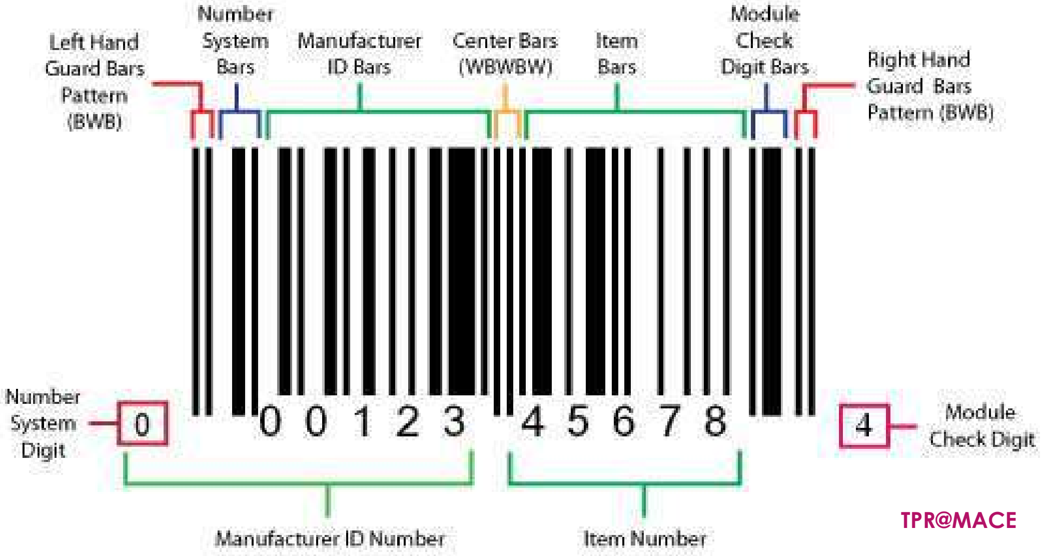


TPR@MACE

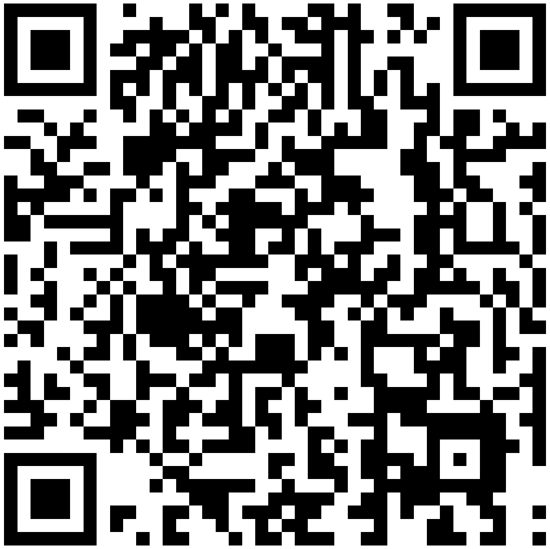
Barcode Readers

- 1D codes, 2D codes
- Optical scanners
- Laser/LED source of light travels to the source of information
- Reflected light sensor + decoder circuit converts optical image into digital data and analyses it
- The output data is a set of 0’s and 1’s which is analysed by a microcontroller


EASY WAY TO LEARN BAR CODE
A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z
1 2 3 4 5 6 7 8 9 0 *



QR Code Readers



- Quick Response Code
- First used in automobile industries
- product identification, tracking, marketing, document management
- Uses standard encoding methods like numeric, alphanumeric…
- Now popular in other industries - reads faster
- QR code – consist of black square dots arranged in a square grid format
- Required data is arranged in both horizontal and vertical components
- Data processed using 'Reed Solomon method'

Motion sensors

- Sensor measures delay b/w successive reflected IR light pulses – IR emitter + phototransistor
- Also, ultrasonic wave echoes can be used to sense the motion of light. Sensor measures delay b/w successive echoes.

Pressure Sensors

- P = Force / m^2
- Pressure transducer, pressure transmitter, pressure sender, pressure indicator.
- Piezometer pressure transmitter – uses a piezo crystal b/w two surfaces. Compression creates electric charges proportional to the pressure applied
- Resistive sensor – measures variation of resistance with force
- Eg: Tyre pressure sensor

Environmental Monitoring Sensor

- Temperature, humidity, barometric pressure, light
- Measurement of the above using various sensors

Location Data

- Determining location of an object
- Using IR sensors, ultrasonic sensors, GPS
- GPS – global positioning system – using satellites

Camera / Image

- Uses CCD (charge coupled device) – has large number of pixels
- It accumulates charge on each of the pixel present
- The accumulation of charges takes place as per the intensity of light
- Colored camera has a set of R, G, B light intensity components at each pixel coordinate

LIDAR

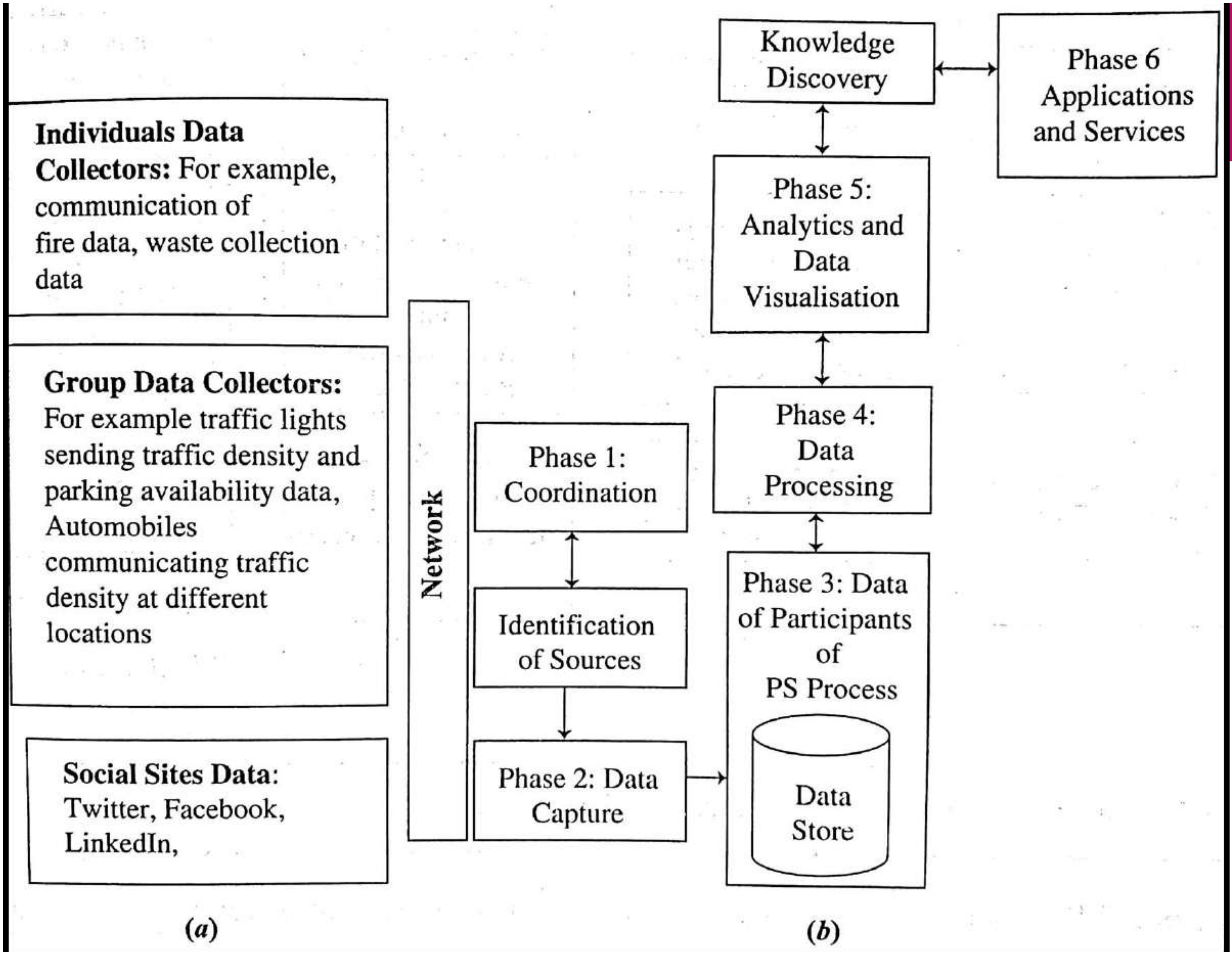
- LIDAR = Light + Radar
- Laser Imaging, detection and ranging]
- Finds distance by throwing light using laser on target
- Sensor senses the reflected light which enables computations of distance.

LASER 3D Imaging

- 3D imaging is feasible using LASER.

Participatory Sensing

- Sensing by the individuals and groups of people contributing sensory information to form a body of knowledge
- Participants – sensors which contribute data Eg: Sensors in a mobile phone – which communicate on the internet on the sensed data with time & date stamps
- Applications – Weather forecasting, environment information, pollutions, waste management, road faults, train management, health of individuals, disaster management



Individuals Data Collectors: For example, communication of fire data, waste collection data

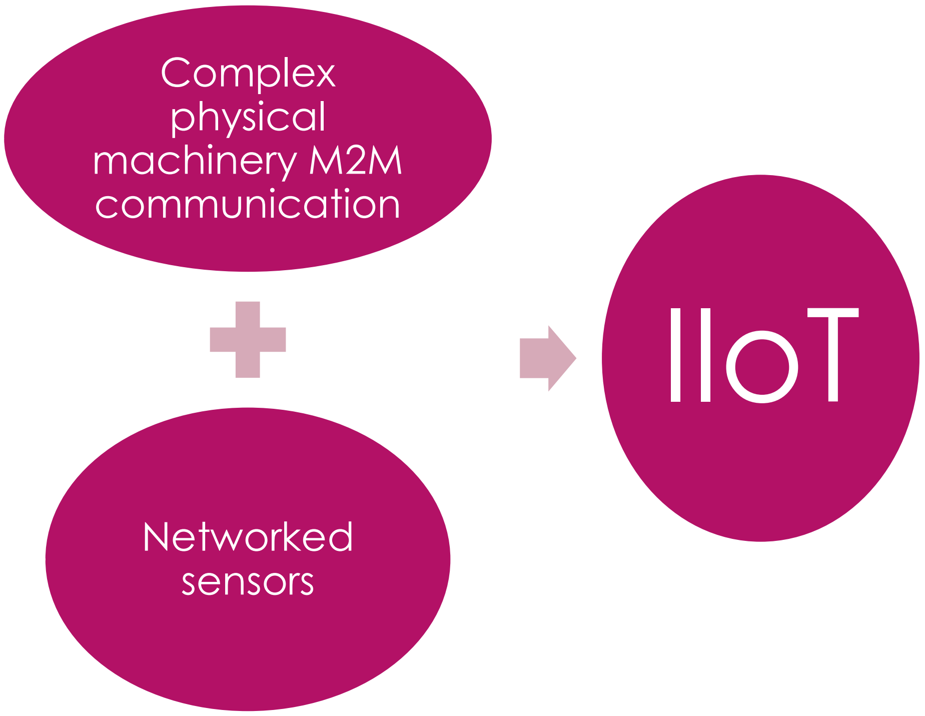
Group Data Collectors: For example traffic lights sending traffic density and parking availability data, Automobiles communicating traffic density at different locations

Social Sites Data: Twitter, Facebook, LinkedIn,

Network
Phase 1: Coordination
Identification of Sources
Phase 2: Data Capture
Phase 3: Data of Participants of PS Process
Data Store
Phase 4: Data Processing
Phase 5: Analytics and Data Visualisation
Knowledge Discovery
Phase 6 Applications and Services
(a)
(b)

IIoT – Industrial IoT

- Use of IoT technology for manufacturing
- Sensors at each stage communicate information on completion – including breakdowns, delays, failures
- Helps company to identify gaps and resolve issues – Optimisation of processes



Complex physical machinery M2M communication

Networked sensors

IIoT

Predictive Maintenance possible

Industrial Internet Consortium

- A body [founded for creation of standards, open interoperability and development of architectures for IIoT

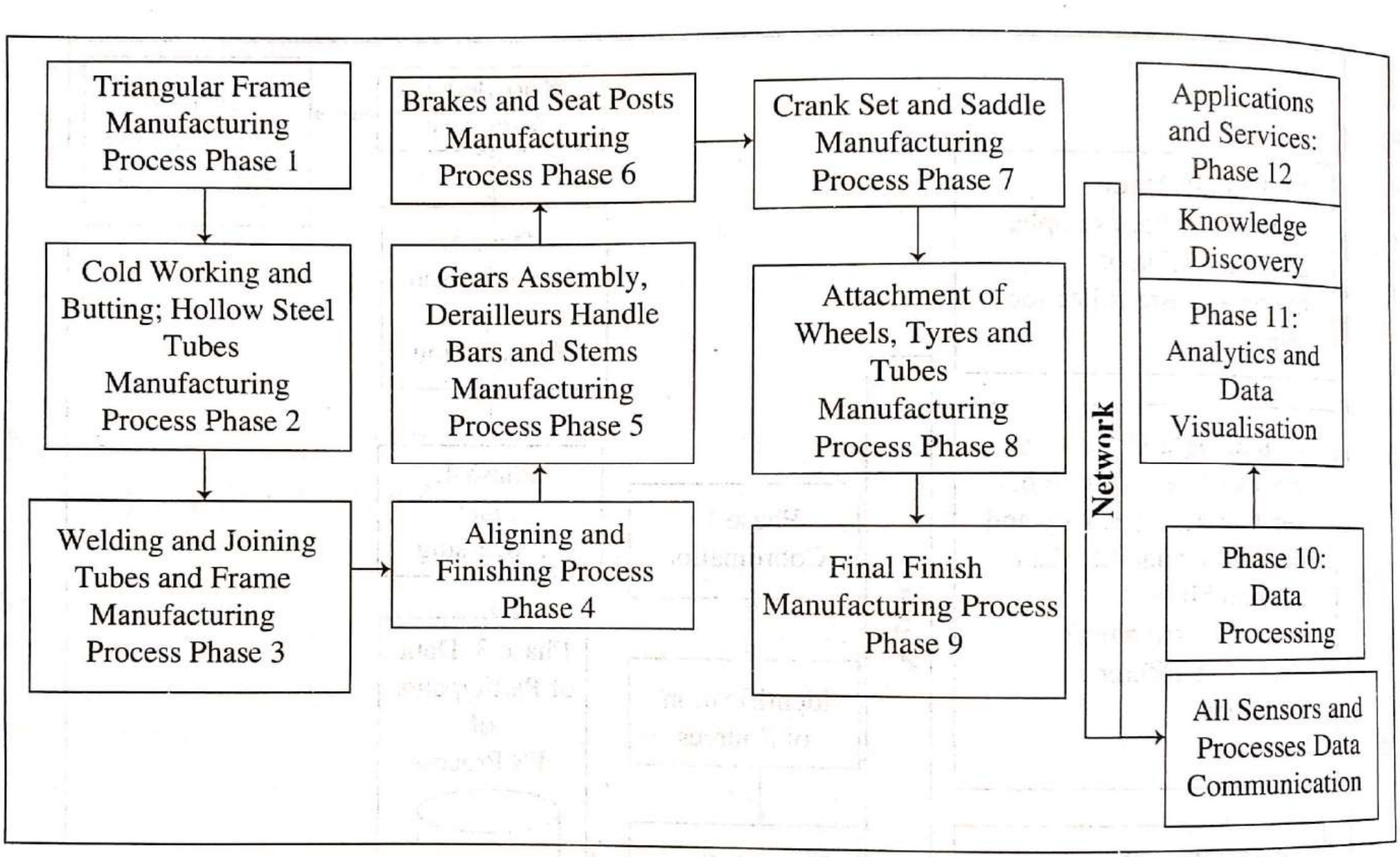

Triangular Frame Manufacturing Process Phase 1
Cold Working and Butting; Hollow Steel Tubes Manufacturing Process Phase 2
Welding and Joining Tubes and Frame Manufacturing Process Phase 3
Aligning and Finishing Process Phase 4
Gears Assembly, Derailleurs Handle Bars and Stems Manufacturing Process Phase 5
Brakes and Seat Posts Manufacturing Process Phase 6
Crank Set and Saddle Manufacturing Process Phase 7
Attachment of Wheels, Tyres and Tubes Manufacturing Process Phase 8
Final Finish Manufacturing Process Phase 9
Network
All Sensors and Processes Data Communication
Phase 10: Data Processing
Phase 11: Analytics and Data Visualisation
Knowledge Discovery
Applications and Services: Phase 12

Figure 7.10 IIoT phases in the bicycle manufacturing process

Automotive IoT

- Enables connected cars, vehicle to infrastructure technology, predictive maintenance of autonomous cars

1. Connected Cars Technology

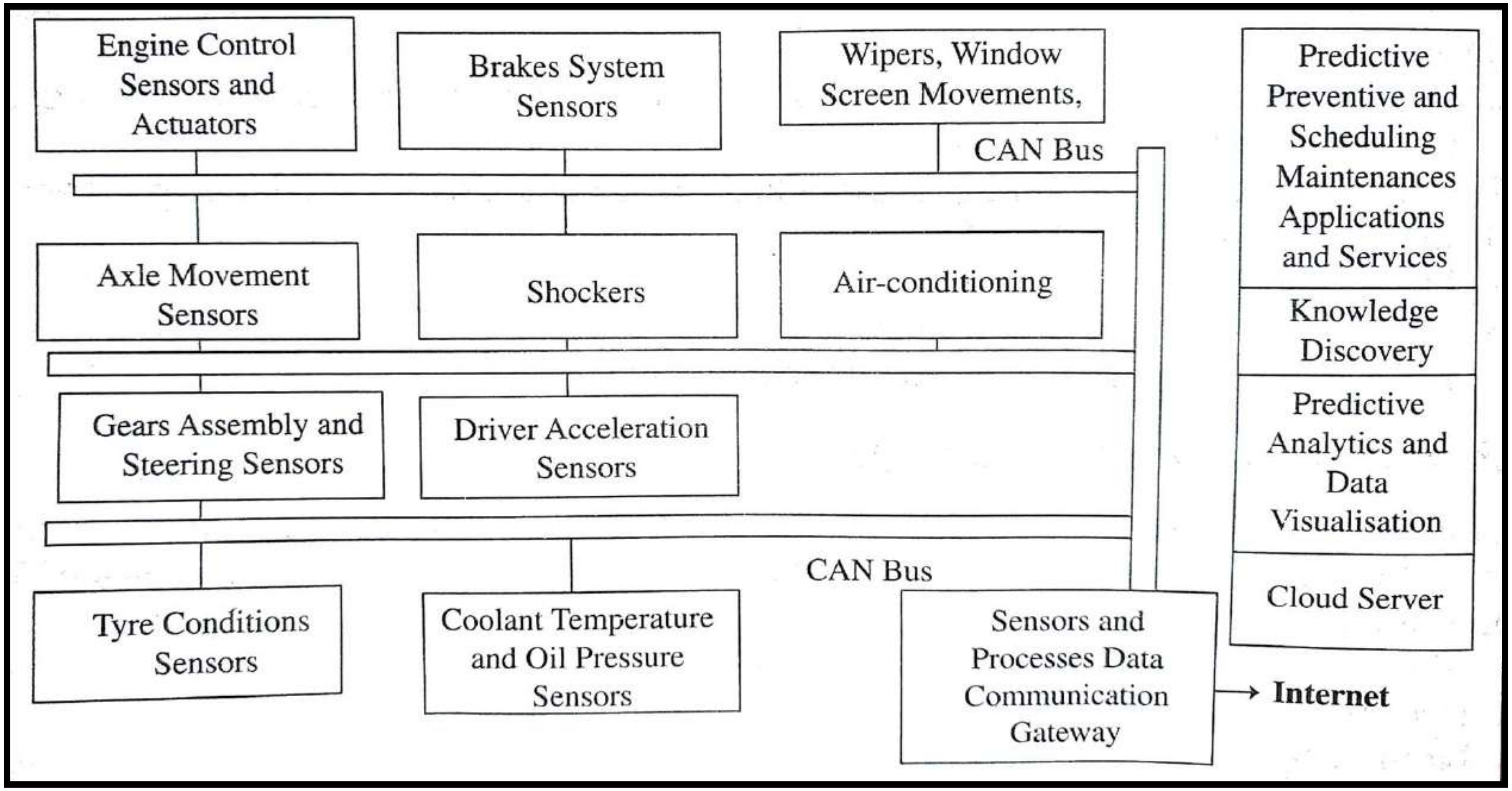
- Display for driver enabling shortest route avoiding congestion
- Customisation of vehicle performance
- Traffic updates
- Theft protection
- Weather updates
- Driver’s health monitoring

2. V2I - Vehicle to infrastructure technology

- Vehicle communicates with other vehicles, surrounding infrastructure and WiFi LAN
- Alerts about accident prone areas
- Vacant parking slot information
- Traffic congestion updates
- Live news, music

Predictive maintenance

- Applications for predictive and preventive maintenance of automobiles


Engine Control Sensors and Actuators
Brakes System Sensors
Wipers, Window Screen Movements,
CAN Bus
Predictive Preventive and Scheduling Maintenances Applications and Services
Axle Movement Sensors
Shockers
Air-conditioning
Knowledge Discovery
Gears Assembly and Steering Sensors
Driver Acceleration Sensors
Predictive Analytics and Data Visualisation
CAN Bus
Cloud Server
Tyre Conditions Sensors
Coolant Temperature and Oil Pressure Sensors
Sensors and Processes Data Communication Gateway
Internet

CAN bus

Controller Area Network

a robust vehicle bus standard designed to allow microcontrollers and devices to communicate with each other in applications without a host computer

Actuators

- A device that takes actions as per the input command
- Motor, speaker, LED or any type of output device converts electrical energy into physical action

LED

Piezo crystals

Speakers

Solenoids

Servomotors

Relay switch

Application of brakes

Ringing an alarm

Turning on / off devices

Varying the control parameter like temperature setting of a room

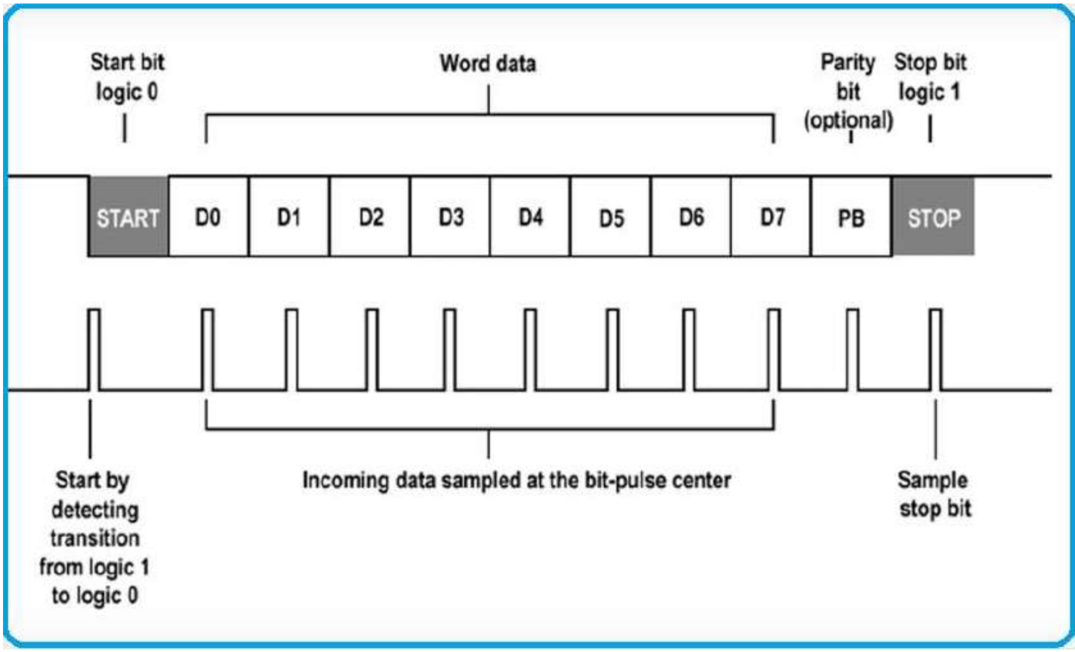
SENSOR DATA COMMUNICATION PROTOCOLS

Serial Bus

- Serial asynchronous communication includes UART
- Serial synchronous interface includes I2C or SPI interfaces

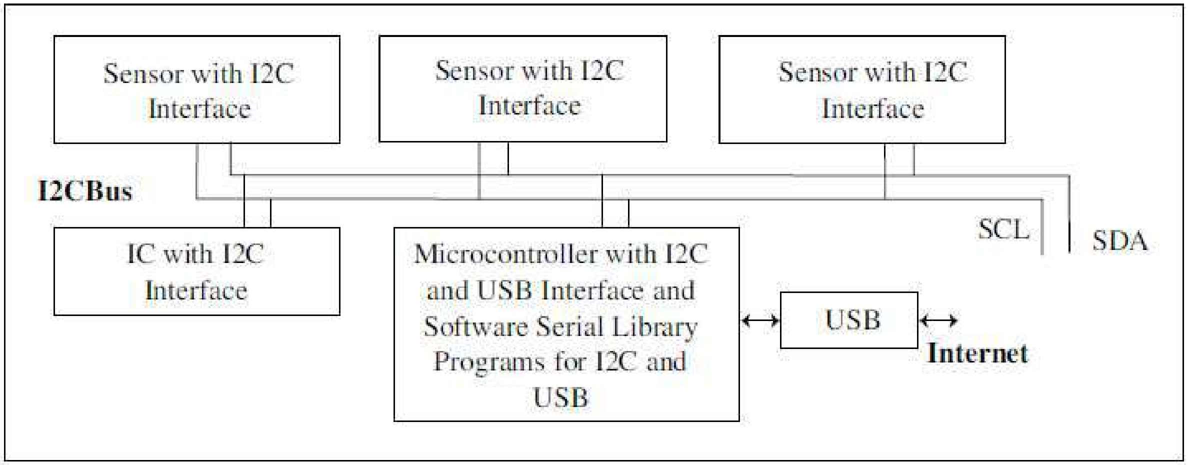
UART

- Universal Asynchronous Receiver/Transmitter
- Sends 8-bit data serially at successive intervals called “**Baud rates**”
- Start bit precedes data. Stop bit succeeds data
- Each character is 8-bit and is coded as per ASCII code
- UART for RFID Tags: A header character is sent before Tag. A tag ID has 10 digit characters. The microcontroller’s IDE Software library contain programs used for reading data from RFID’s using UART.



I2C

- Inter-Integrated Circuit and is pronounced I-squared-C and is alternatively known as I2C or IIC
- I2C is synchronous, so the output of bits (serial data-SDA) is synchronized to the sampling of bits by a clock signal (serial clock-SCL) shared between the master and the slave. The clock signal is always controlled by the master.
- Industrial 100kbps I2C ; Industrial 400kbps I2C


Sensor with I2C Interface
Sensor with I2C Interface
Sensor with I2C Interface
I2CBus
SCL
SDA
IC with I2C Interface
Microcontroller with I2C and USB Interface and Software Serial Library Programs for I2C and USB
USB
Internet

Bus SCL and SDA lines for serial synchronous data communication using I2C protocol

TPR@MACE

LIN

- Local Interconnect Network
- serial bus network protocol for communication between automobile circuits, sensors and actuator circuits, components and systems
- Eg: window movements, seat movements and wipers.
- The protocol is simpler to use compared to Controller Area Network (CAN bus) in automobiles.
- LIN communication is single master with maximum 15 slaves

CAN

- Controller Area Network **(CAN bus)**
- Different controller circuits communicates using the same set of wires
- The controllers provide the controls for brakes, engine, electric power, lamps, temperature, air conditioning, cargate, front display panel, meter display panels and cruising.
- Medical electronics and industrial-plant serial communication also use the CAN bus.

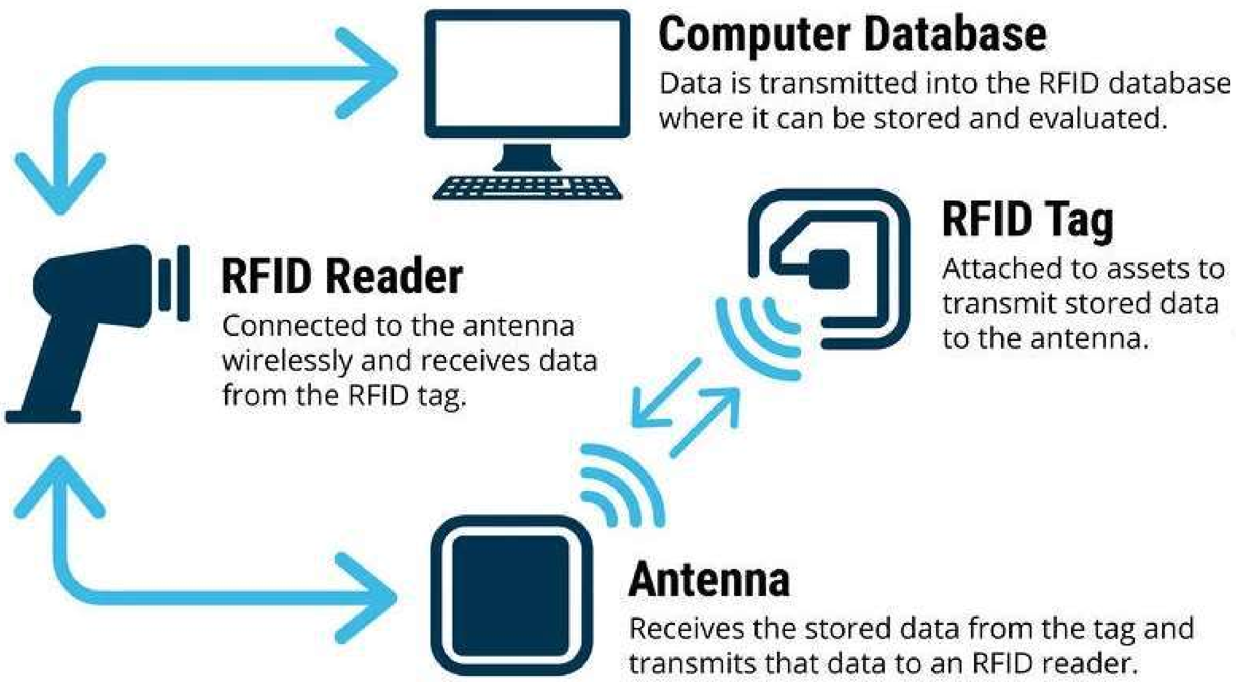
LIN bus vs CAN bus

- LIN is lower cost (less harness, no license fee, cheap nodes)
- CAN uses twisted shielded dual wires 5V vs LIN single wire 12V
- LIN is deterministic, not event driven (i.e. no bus arbitration)
- LIN clusters have a single master - CAN can have multiple
- CAN offers up to 1 Mbit/s vs. LIN at max 20 kbit/s

USB

- Universal Serial Bus (**USB**) is an industry standard that establishes specifications for cables and connectors and protocols for connection, communication and power supply (interfacing) between computers, peripherals and other computers.
- Both host and serial devices can function in a system
- Maximum 127 devices can connect to a host.
- Three standard protocols for USB are:
  - >USB 1.1 (a low speed 1.5 Mbps 3 meter channel along with a high speed 12 Mbps 25 meter channel)
  - >USB 2.0 (high speed 480 Mbps 25 meter channel) and wireless USB (high speed 480 Mbps 3 m).
  - >USB 3.1 Gen 1 – SuperSpeed, 5 Gbit/s
  - >USB 3.1 Gen 2 – SuperSpeed+, new 10 Gbit/s

RFID TECHNOLOGY

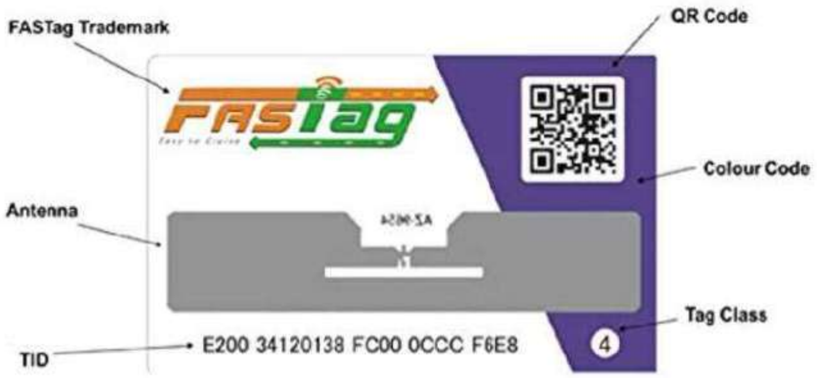


Computer Database
Data is transmitted into the RFID database where it can be stored and evaluated.

RFID Reader
Connected to the antenna wirelessly and receives data from the RFID tag.

RFID Tag
Attached to assets to transmit stored data to the antenna.

Antenna
Receives the stored data from the tag and transmits that data to an RFID reader.



The reader circuit of an ID can use UART or NFC protocol to identify the tag, when the RFID tag is at a distance less than 20 cm. An active NFC device/mobile generates an RF field which induces the currents in RFID and generates enough power for RFID. Using that power, the RFID transmits the identification of tag contents to a reader. Is is then transmitted along with the additional information to a remote server or cloud connected through the Internet. An RFID tag has an advantage over a barcode or QR code in terms of simpler processing of the RFID data. It can also be made invisible to a person. This is because it uses shortrange RF transceivers instead of light or laser.

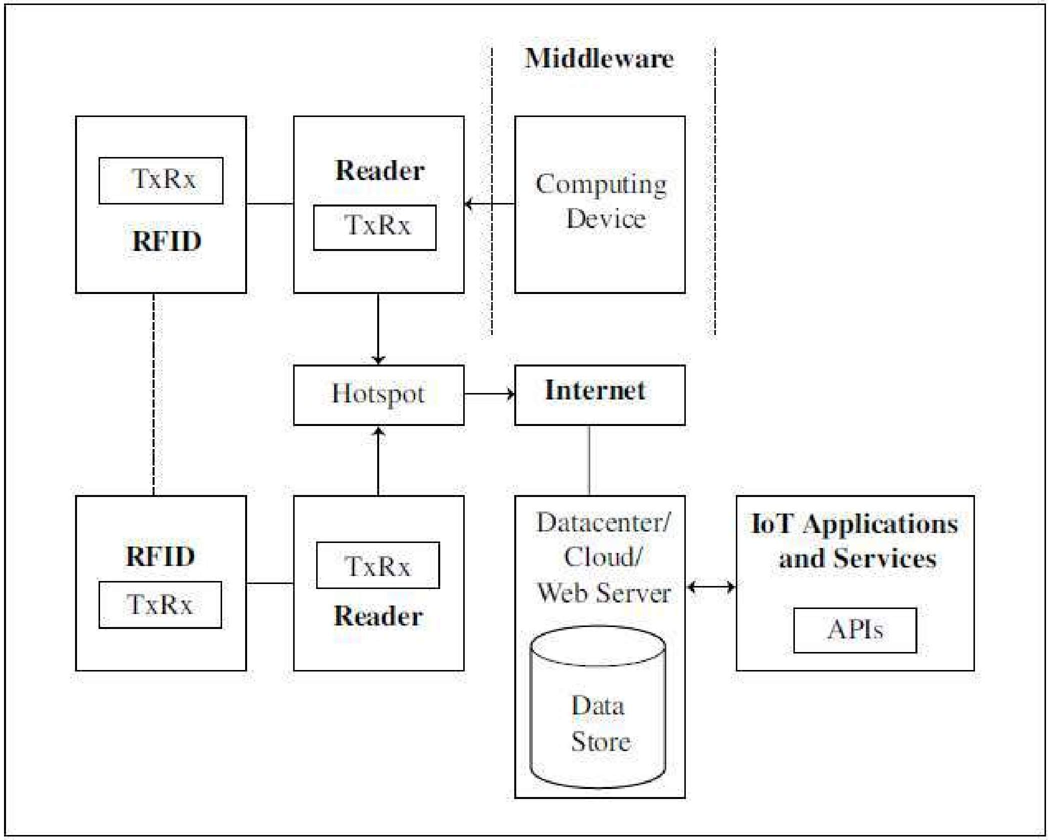
Components of an RFID System

1. RFID is a tiny chip which functions as a tag or label onto an object. The chip is one of three types—passive, active and battery powered passive (battery switches when reader is nearby). A transceiver is in-built at the chip. It communicates in a range 10 cm to 200 m according to the chip. The chip does UART communication to the reader either using RF link or does NFC.
Standard freq ranges include 20 kHz to 150 kHz, 13.56 MHz, 433 MHz, higher when using UHF and microwave frequencies.

2. Data processing subsystem: A reader associates a data processing subsystem which consists of a computing device and a middleware and provides connectivity to the Internet, directly or through a gateway which includes a data adaptation sublayer. Example of a reader is SparkFun SEN-08419 for prototype developments.

3. Middleware: Middleware are software components used a the reader, read manager, data store for the transaction data store and APIs of the applications.

4. Applications and services and other associated applicatior software use the data store at the cloud or web server.



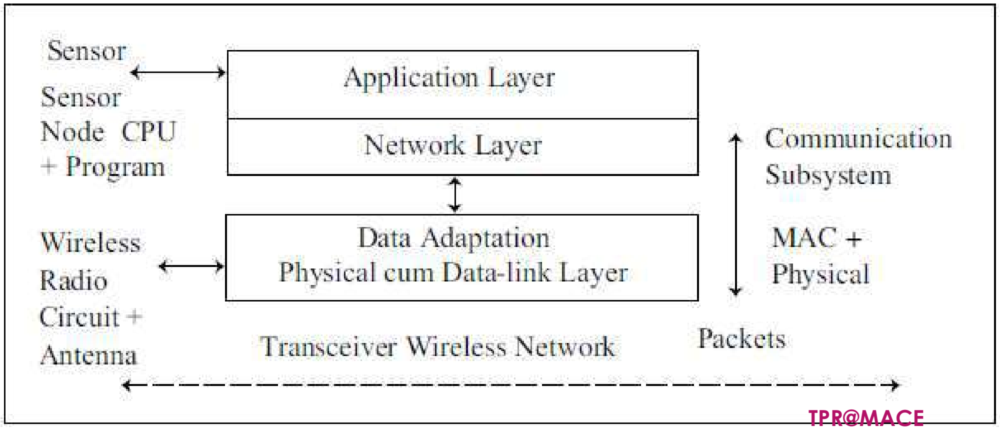
Components needed in a system for RFID IoT applications and services

WSN – WIRELESS SENSORY NETWORKS

- Def: WSN is defined as a network in which each sensor node connects wirelessly and has the capability of computation, for data compaction, aggregation and analysis. Each one also has communication as well as networking capabilities. A WSN consists of spatially distributed autonomous devices (sensors).
- A set of sensors can be networked using a wireless system.
- They cooperatively monitor the physical and environmental conditions, such as temperature, sound, vibration, pressure, motion, or hazardous gas-leaks and pollutants at different locations
- Each node of the WSN has an RF transceiver. The transceiver functions as both, a transmitter and receiver.

WSN Node Architecture

Three-layer architecture of a node. The three layers are application layer, network layer , physical cum data-link layer.
Two basic architectures for networking of the nodes are layered architecture and multicluster architecture.



LAYERED ARCHITECTURE
AND
MULTICLUSTER ARCHITECTURE

Prototyping the embedded devices for IoT/M2M

Embedded Device

- A device which embeds software into a computing platform and performs computations and communication for specific systems

In IoT/M2M – devices generate data – ie embedded devices, sensors, systems at the physical layer

This data needs computations at a data adaptation gateway – enriched data communicates through internet for analytics, visualization, knowledge discovery, applications and service

MCU

- Microcontroller unit means a single chip VLSI unithaving limited computing capabilities – has memory, i/o devices etc…

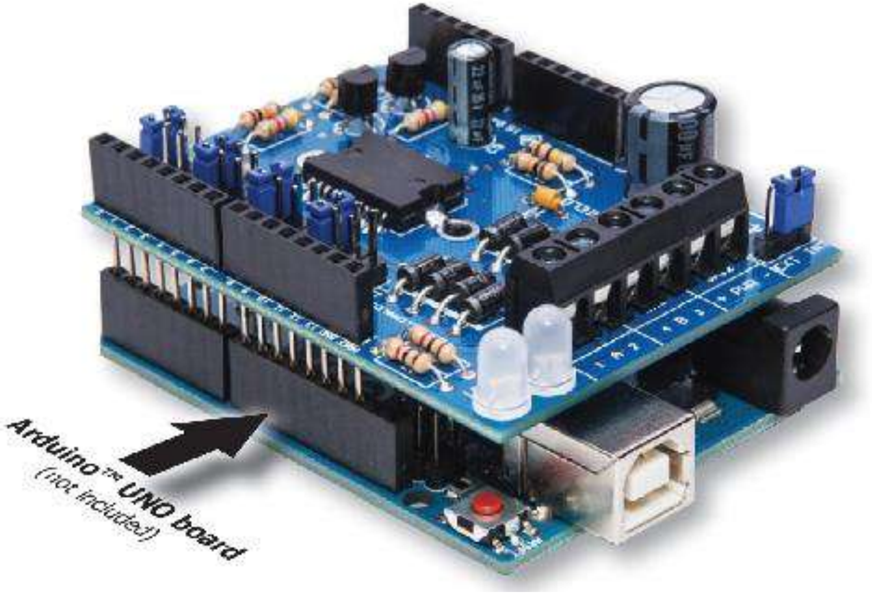
GPIO - General purpose Input Output Pins

- Pins that can be used apart from I/O operations – like Rx, Tx, PWM, analog I/P, analog O/P etc…

Board – hardware with an MCU and connectors, battery, regulators …

Platform

- A set consisting of computing and communication hardware, software and OS
- Platform helps in working with different software, API's, IDE and middlewares



Module

- a small modular hardware which can be placed on the board
- Eg: RF module

Shield

- Supporting circuit with connection pins and sockets and supporting software
- Helps connect the board to an external circuits. EG: Arduino Shields
- They provide extra features like connectivity to Zigbee, Bluetooth, WiFi, GSM etc…

Inperrupt

- An action in which, a running program interrupts a hardware signal

IDE

- a set of software components and modules which provide a software environment for developing and prototypting

Embedded Computing

1. Embedded Software

- Software consists of instructions, commands, data
- Software does the bootloading and enables appplications & services
- Software includes an OS, device API's and middleware
- **<u>Bootloader</u>**: a program which runs at the start of a computing device such as MCU.
  >It initiates loading of system OS.
  >It facilitates use of system hardware and n/w capabilities
  >bootloading is complete when normal operational run-time environment is reached

1. Embedded Software….contd…

- **Operating System**
- An operating system facilitates use of system hardware and networking capabilities.
- After OS loads into RAM, then, MCU starts normal operational run-time environment.
- OS controls multiple processes and device functions
- OS enables memory allocation to various processes ; prioritizes processes ; enables use of network hardware & device hardware functions ; enables execution of software components
- Eg: Linux, Arduino Linux – runs on Arduino circuit boards
- **Real-Time OS**
- RTOS is an OS that enables real-time execution of processes on computing and communication hardware.
- RTOS prioritizes & priority allocation concepts.
- LynxOS ; OSE ; QNX ; RTLinux ; VxWorks ; Windows CE

2. Integrated Development Environment

- A set of software components and modules which provide the software and hardware environment for developing and prototyping.
- Enables code development on a computer
- IDE consist of > device API's, libraries, compilers, RTOS, simulator, editor, assembler, debugger, emulators, logic analyser, EPROM, EEPROM etc…
- Eg: Arduino has an open source IDE on Arduino website
- IDE enables a prototype design – to develop embedded hardware and software platforms, simulating and debugging

- **Simulator**

- Software that enables development on the computer without any hardware

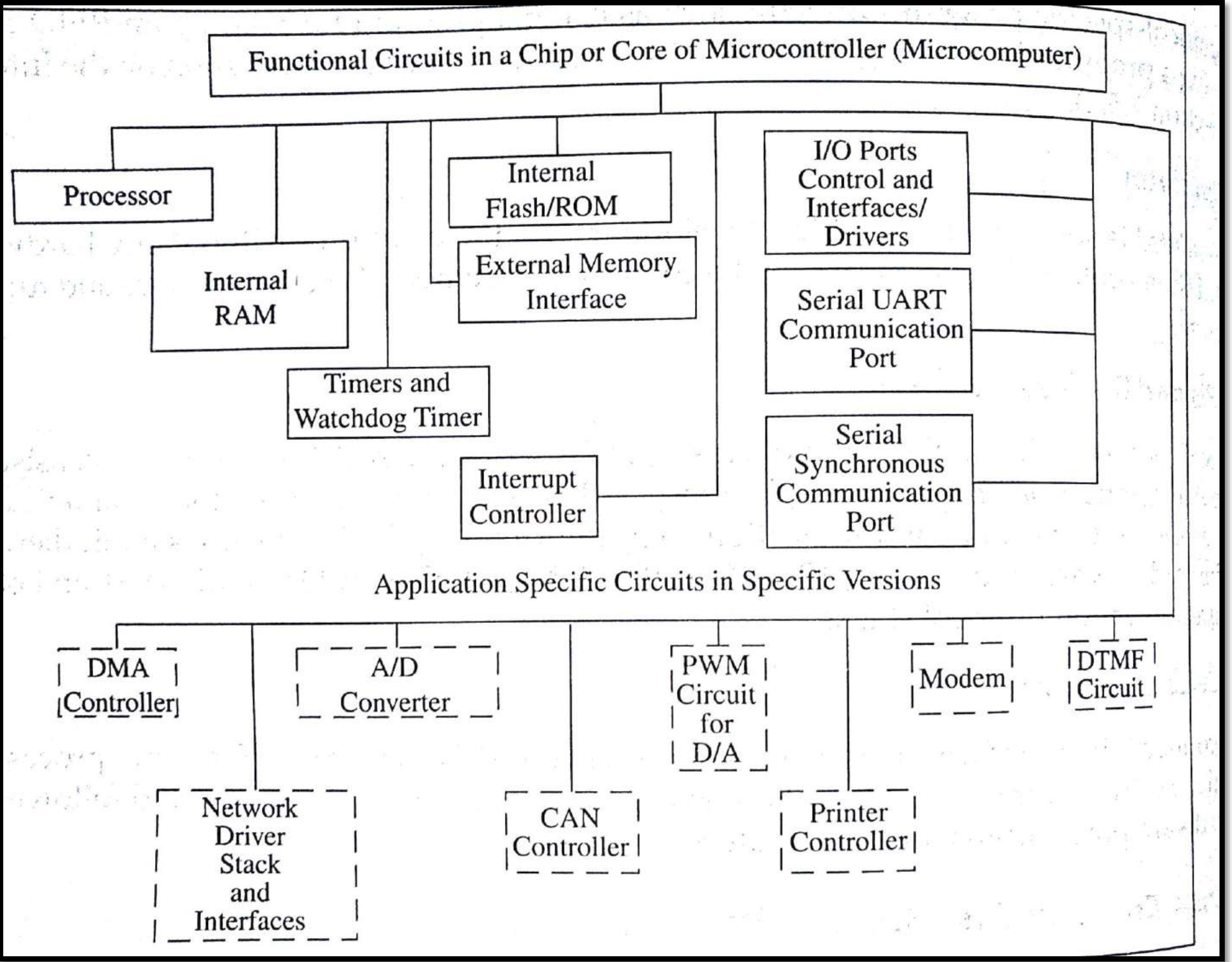
- **API**
- **Device Interfaces**

3. Embedded Hardware Units

- Includes the following:
- 1. Single VLSI chip
- 2. ASIP – Application specific Instruction set Processor (MCU)
- 3.ASIC – Application specific IC
- 4. SoC – System on chip with SD card for embedded software & OS

- **<u>MCU</u>**

- Single chip VLSI unit – having enhanced i/o capabilities – having on chip functional units like RAM,flash,IO ports,GPIO, serial interfcaes, timers etc…
- Application specific MCU has additional functional units like PWM circuits, ADC etc…


Functional Circuits in a Chip or Core of Microcontroller (Microcomputer)
Processor
Internal RAM
Timers and Watchdog Timer
Internal Flash/ROM
External Memory Interface
Interrupt Controller
I/O Ports Control and Interfaces/ Drivers
Serial UART Communication Port
Serial Synchronous Communication Port
Application Specific Circuits in Specific Versions
DMA Controller
A/D Converter
PWM Circuit for D/A
Modem
DTMF Circuit
Network Driver Stack and Interfaces
CAN Controller
Printer Controller

- 8 bit, 16 bit 32 bit…
- Clock freq – 8MHz, 16, 100 or 200
- RAM – 4kB, 16, 32, etc…
- EEPROM & flash memory – 512B , 1 kB, to 512kB
- MCU may have timers, i/o ports, gpio pins, serial pins, ADC, PWM, I2C, LCD interface, Zigbee interface, Ethernet, Modem etc…

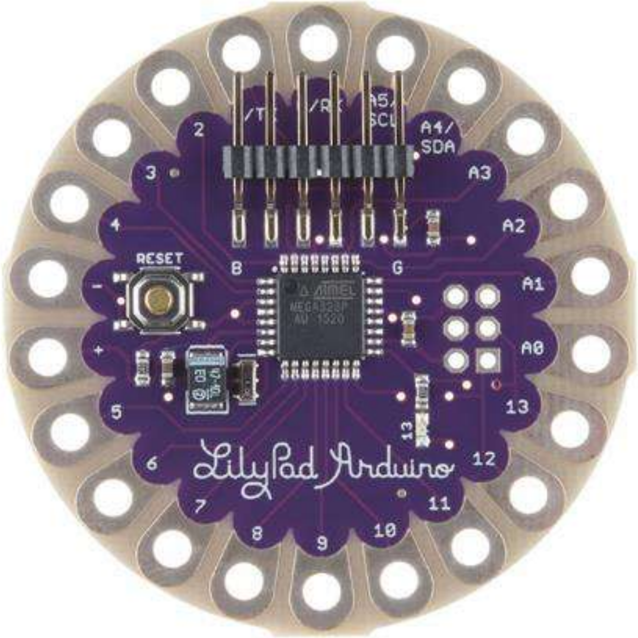
3. Embedded Hardware Units contd…

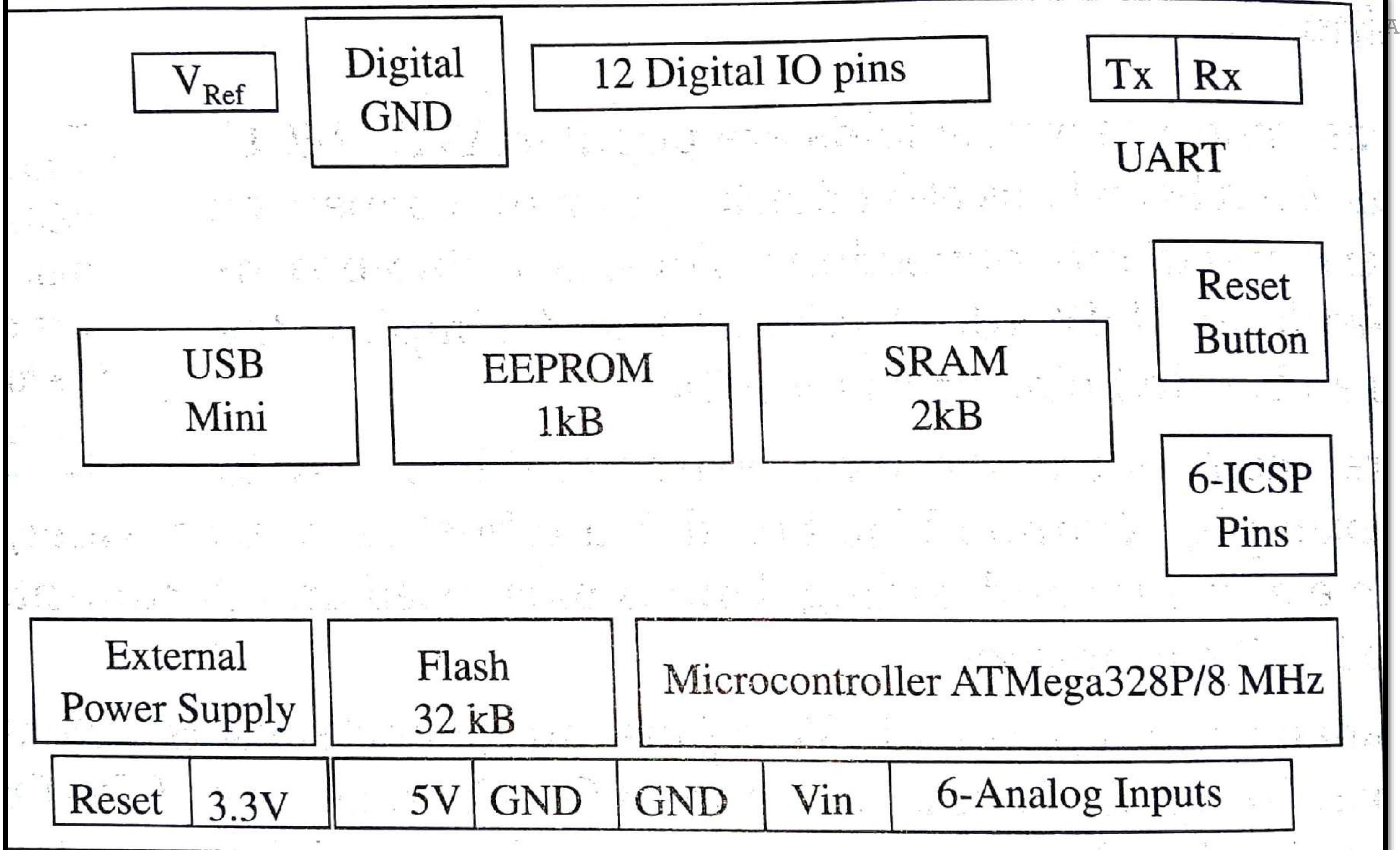
- **<u>System on Chip</u>**
- Complex embedded systems like that in a smart phone will be designed to a single silicon chip – having multiple processors, hardware units and softwares, digital & analog circuits
- SoC = system on a VLSI chip
- **<u>How to select an embedded platform?</u>**
- Price > Open Source availability > ease of app development > performance required …
- <u>Hardware:</u> Processor speed, RAM, Connection to Zigbee, Bluetooth, WiFi, USB host, Sensor+Actuator interfaces, Sensor communication , power requirement…
- <u>Software platforms :</u> Open source availability, IDE with device API, libraries, OS/RTOS, emulator, simulator, internet protocols, cloud & sensor cloud platforms, data storage & services

Popular Embedded Platform boards, modules and supporting circuits

ARDUINO

- IDE is open source
- Arduino Uno is a very popular board from Arduino
- Analog I/P pins and PWM can connect sensors, actuators and analog circuits.
- Digital pins can connect on/off states-inputs from sensors and outputs to actuators
- A board with shields inserted makes wireless connection to Zigbee, Bluetooth, wifi, gsm, RF modules
- Development boards for IoT devices– Arduino Ethernet ; Arduino WiFi ; Arduino GSM shields
- Development boards for wearable devices devices– Arduino Gemma, Lilypad



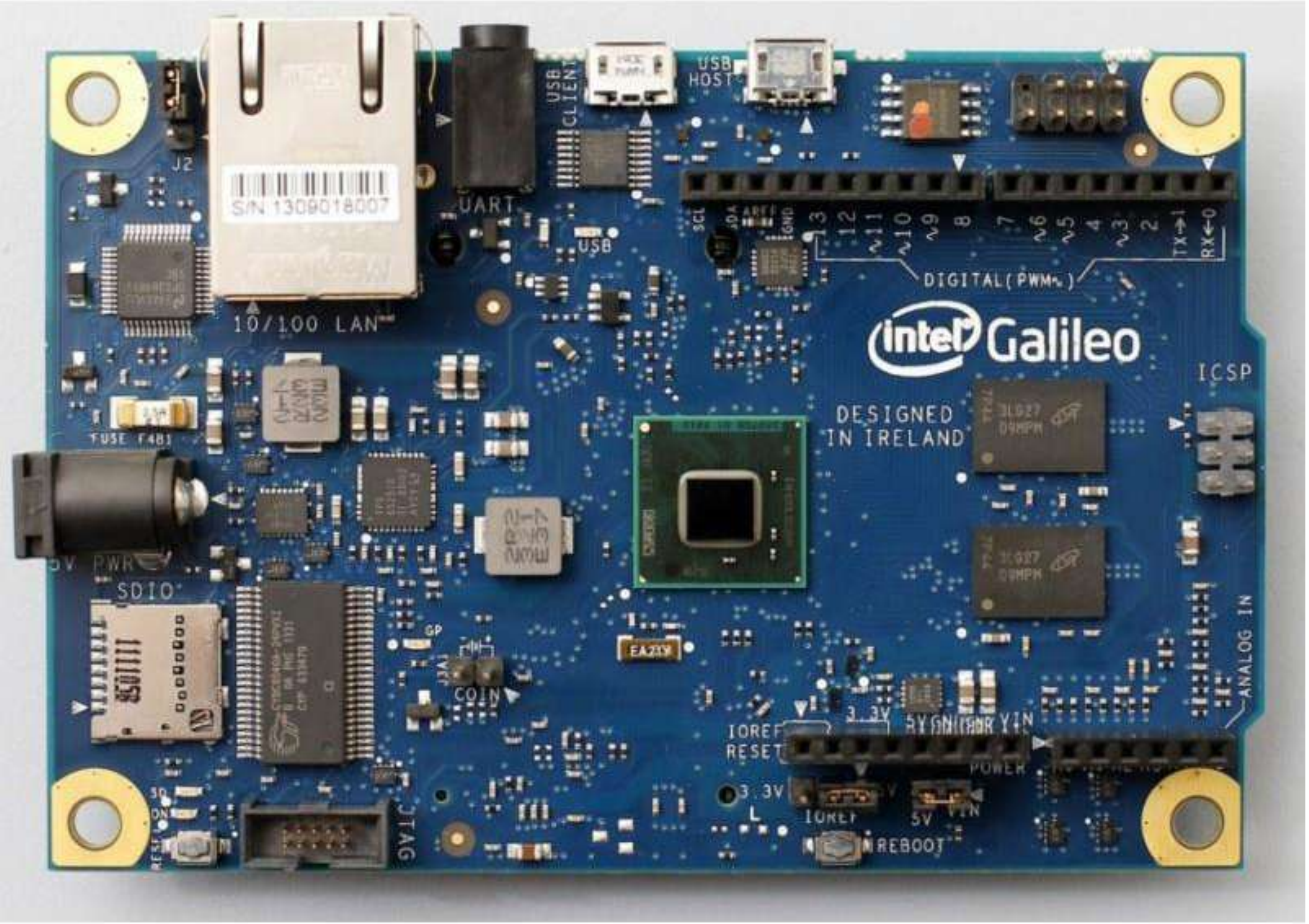

V_{Ref}
Digital GND
12 Digital IO pins
Tx
Rx
UART
Reset Button
USB Mini
EEPROM 1kB
SRAM 2kB
6-ICSP Pins
External Power Supply
Flash 32 kB
Microcontroller ATMega328P/8 MHz
Reset
3.3V
5V
GND
GND
Vin
6-Analog Inputs

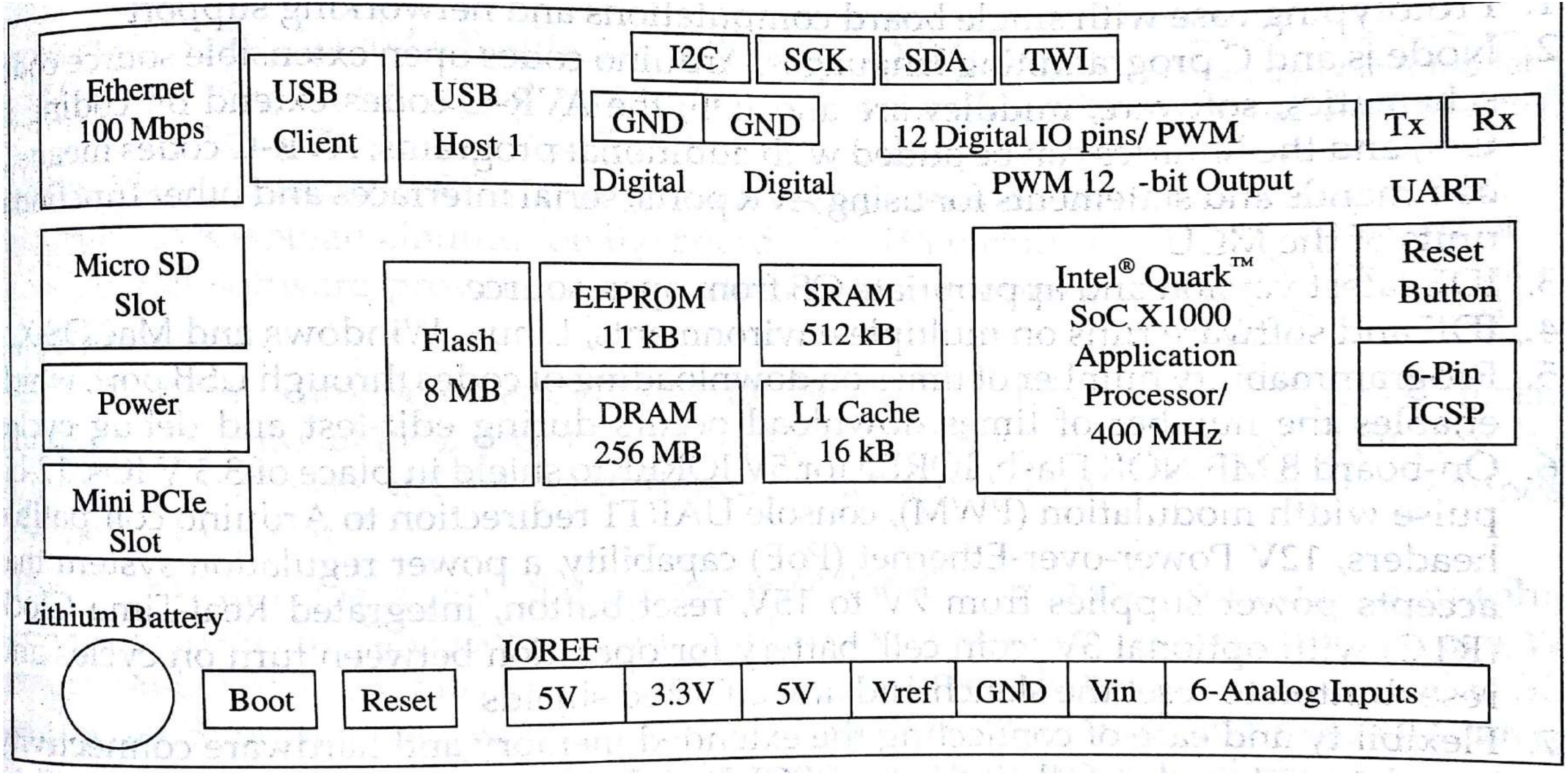
Arduino contd…

- IoT application – where device doesn’t require intensive computing/graphics
- Arduino Uno – a reference model for Arduino platform – an MCU board – has USB connection – after development in IDE on a computer, the program can be burned into Uno boards
- <u>*General Advantage of Arduino:*</u>
  - Prototyping ease
  - Flexibility in assembling the board
  - Open extensible source codes, software, middleware and IDE
  - Coding derived from C++
  - IDE runs on Linux, windows and Mac OS
  - Hardware is open source – extensible using modules, shields…

Intel Galileo

- Based on intel Pentium architecture – single threaded, single core, 400Mhz constant speed processor
- Eg: Intel Quark SoC X1000
- Galileo is Arduino certified. Ie , its compatible with shields designed for Arduino uno and Arduino IDE
- Galileo supports max of 30 sensors and accessories for Arduino
- ***IoT applications:***
  - Making everything smart… - health monitoring, fitness devices, watches, sensors, cameras
  - Codes Can be developed on a PC run IDE in linux/windowsmac
  - Architecture….


J2
S/N 1309018007
UART
USB CLIENT
USB HOST
USB
10/100 LAN
SCL
SDA
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
intel Galileo
ICSP
DESIGNED IN IRELAND
FUSE F481
5V PWR
SDIO
GP
COIN
IOREF
RESET
3.3V
5V GND GND VIN
POWER
ANALOG IN
JTAG
3.3V
IOREF
5V
VIN
REBOOT


Ethernet 100 Mbps
USB Client
USB Host 1
I2C
SCK
SDA
TWI
GND
GND
Digital
Digital
12 Digital IO pins/ PWM
PWM 12 -bit Output
Tx
Rx
UART
Micro SD Slot
Power
Mini PCIe Slot
Flash 8 MB
EEPROM 11 kB
SRAM 512 kB
DRAM 256 MB
L1 Cache 16 kB
Intel® Quark™ SoC X1000 Application Processor/ 400 MHz
Reset Button
6-Pin ICSP
Lithium Battery
Boot
Reset
IOREF
5V
3.3V
5V
Vref
GND
Vin
6-Analog Inputs

Features of galileo

- Prototyping ease
- C programming compatibility – Arduino open codes
- IDE is Open Source – Arduino IDE
- On board 8Mb NOR flash
- 12 bit PWM
- 12V Power over Ethernet capability
- Power regulation – 7 V to 15V
- Reset button
- Realtime clock – with 3V coin cell
- Flexible – connecting the extended memory through PCI, micro SD slot etc…

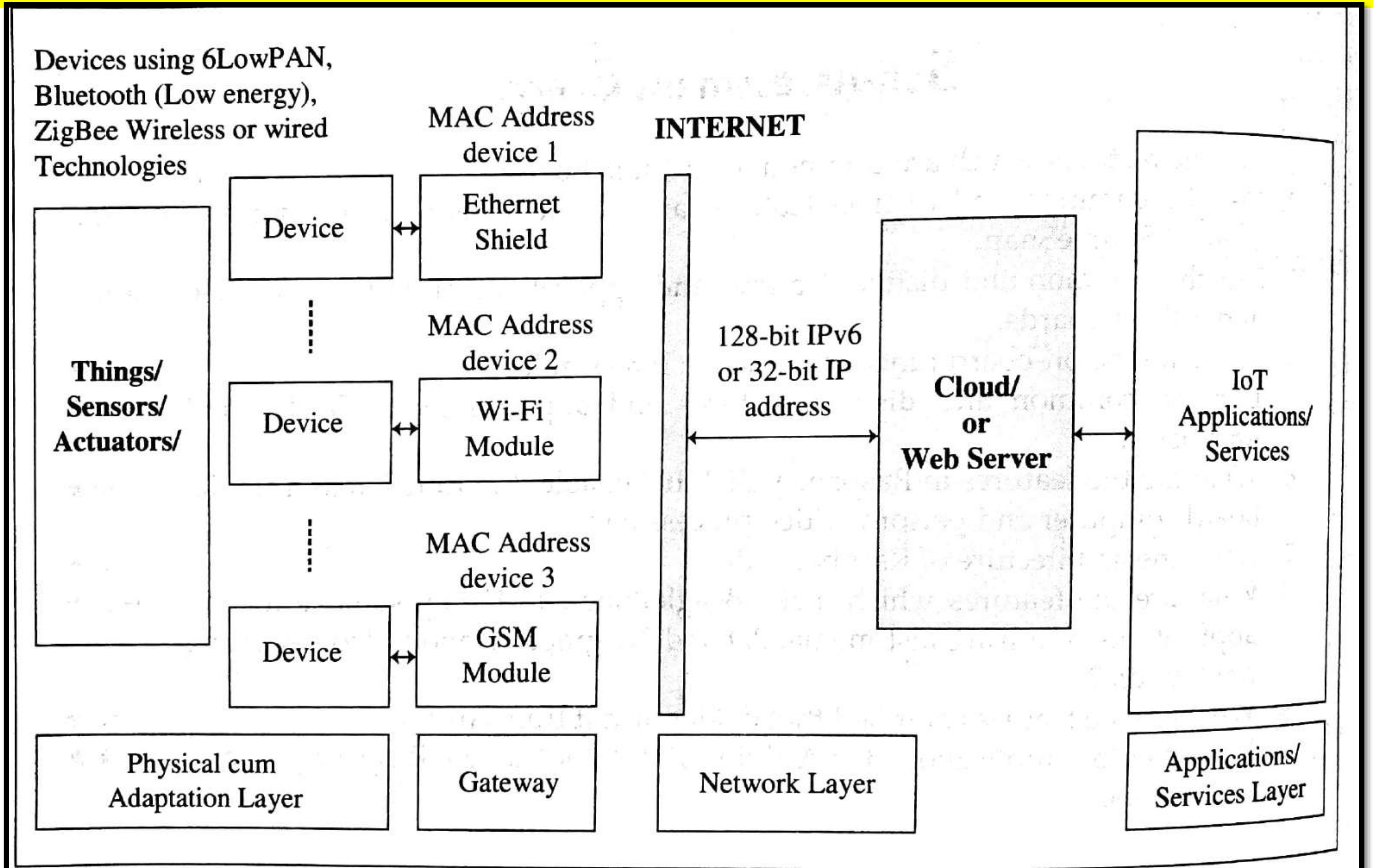
Intel Edison

Raspberry Pi

Beaglebone

mBed

Connecting Things to Cloud / Internet


Devices using 6LowPAN, Bluetooth (Low energy), ZigBee Wireless or wired Technologies
Things/ Sensors/ Actuators/
Device
Device
Device
MAC Address device 1
Ethernet Shield
MAC Address device 2
Wi-Fi Module
MAC Address device 3
GSM Module
INTERNET
128-bit IPv6 or 32-bit IP address
Cloud/ or Web Server
IoT Applications/ Services
Physical cum Adaptation Layer
Gateway
Network Layer
Applications/ Services Layer

Connecting things to the cloud

- Things first connect to embedded computing devices
- Data adapts at this level for communication
- The device may be Arduino, galileo, Rpi etc…
- Each connected device on LAN has a MAC address (media access control address)
- Devices communicate using the Bluetooth, zigbee or other wireless/ wired methods
- When device has to communicate over internet, they use 32 bit IP address or 128 bit IPv6 address
- RARP - Reverse Address Resolution Protocol - translates MAC address to IP
- ARP - Address Resolution Protocol - translates IP address to MAC

Connecting Arduino USB to internet

- Arduino board IDE supports USB
- Using IDE USB port function, the device is connected to mobile or PC
- PC / mobile is then connected to internet using network interface cards

Arduino to internet

- Arduino IDE supports Ethernet protocol library
- Ethernet LAN connects to network router directly or through WiFi

Arduino to Wifi

- Arduino IDE supports WiShield Lirary
- Wishield connects to a network router
- However, sufficient energy is required for connection

DESIGN LAYERS AND PHASES

Design Layers – Different IoT architecture need different number of layers for developing the product.

For eg: simple systems like a smart umbrella will need designing the following layers:

- Layer 1 – physical objects/sensors
- Layer 2 – intranet, internet, mobile service provider
- Layer 3 – controller/monitor

For complex processes, we will need a standard model as described in architectures:

IoT Architecture	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6
Oracle	Gather	Enrich+ Stream	Manage	Acquire	Analyse + Intelligence	---
IBM	Gather	Consolidate	Connect + Collect	Assemble	Manage, Analyse	Enterprise integration and application

Design Phases

Phases in a participatory n/w IoT

- Phase 1 – gather data from - individual / collective groups of sensors or from social media sorces
- Phase 2 – data capture
- Phase 3 – consolidation
- Phase 4 – connect, data processing, validation
- Phase 5 – Analytics + visualization
- Phase 6 – application, services
- Phase 7 – application integration

Design Complexity

Level 1

- Eg: internet of umbrella
- Consists of a single physical object and depends on data from a single source (weather station)

Level 2

- Eg: internet of street lights
- Group of streetlights connected to a group controller through intranet
- Group controller coordinates and connects with internet & applications, services
- Uses cloud/server database platform for acquiring, organizing data

Level 3

- Eg: internet of RFID’s, internet of ATM’s
- It provides for a number of tasks like tracking, security, inventory, control & supply chain management
- Level 3 uses cloud / server platform – acquires and organizes enriched data points, event triggers, alerts at a database, analyses the data with analytics and visualizes it

Level 4

- Eg: ‘Internet of WSN’s , smart homes, city systems
- Requires server platform for acquire, organize, analyse, visualize steps of multiple sources of data and connects to a no of networks.
- ‘internet of rail track sensors – in rail track fault prediction
- ‘internet of oil pipeline sensors
- Internet of weather information – pollution, waste management, road faults…
- Uses networked devices , coordinators, and centralized server cloud platform

Level 5

- Eg: Internet of Automotive components and predictive automotive maintenance application and service
- Uses multi input data sources and a cloud platform
- System cloud acquires, organizes performs data , events, triggers and streams
- System extracts intelligence and performs knowledge discovery & management

Level 6

- Eg: Industrial IIoT
- They analyse data points, triggers, events and alerts from WSN's
- Eg: Anomaly detection systems, production & value added services
- They use multi-input data, multi server cloud platform. System cloud acquires, organizes performs data , events, triggers and streams
- System extracts intelligence, deploys machine learning and performs knowledge discovery & management

Complexity	Features	Examples
Level 1	• Single physical objects and web services • Doesn't use cloud / server platform	• Smart umbrella
Level 2	• Group controllers • Central coordinating server • Uses cloud / server database platform	• Smart Streetlight control • Smart drip irrigation
Level 3	• Provisions many applications • Uses cloud / server database platform • Platform acquires & organizes enriched data, event triggers, alerts, analyses the data with analytics and visualizes the analyzed data	• Internet of RFID's & its applications
Level 4	• Use networked devices • Group controllers • Central coordinating server • Platform acquires & organizes enriched data, event triggers, alerts, analyses the data with analytics and visualizes the analyzed data - & processes intelligence • Provisions many applications	• Internet of WSN's – Internet of rail track sensors, oil pipeline sensors etc…

Complexity	Features	Examples
Level 5	• Uses multi-input data sources on Cloud platform • Platform acquires & organizes enriched data, event triggers, alerts, analytics and visualization • Extracts intelligence • Deploys machine learning and perform knowledge discovery and management	• Automated components and predictive automotive maintenance application
Level 6	• Integration of complex physical machinery M2M and IoT communication • Platform acquires & analyses data points, triggers, events, alerts from networked sensors and multiple data sources • Multi-input data sources, multi-server cloud platforms, • Extracts intelligence, Deploys machine learning and perform knowledge discovery and management • Provisions for numerous applications	• Industrial manufacturing Processes • Production/manufacture driven value chain

Some examples of Tools, Projects, Platforms available for IoT prototyping, development and deployment

Eclipse IOT-

- most popular Java IDE
- Excellent UI's, windows builder, integration with XML editor
- plugins enable development in C, C++, python …

Oracle IoT –

- refers to provisioning of Oracle Java embedded, event processing solutions
- facilitates seamless communication b/w all elements of IoT architecture
- Also has access to Oracle PaaS cloud

KaaIoT –

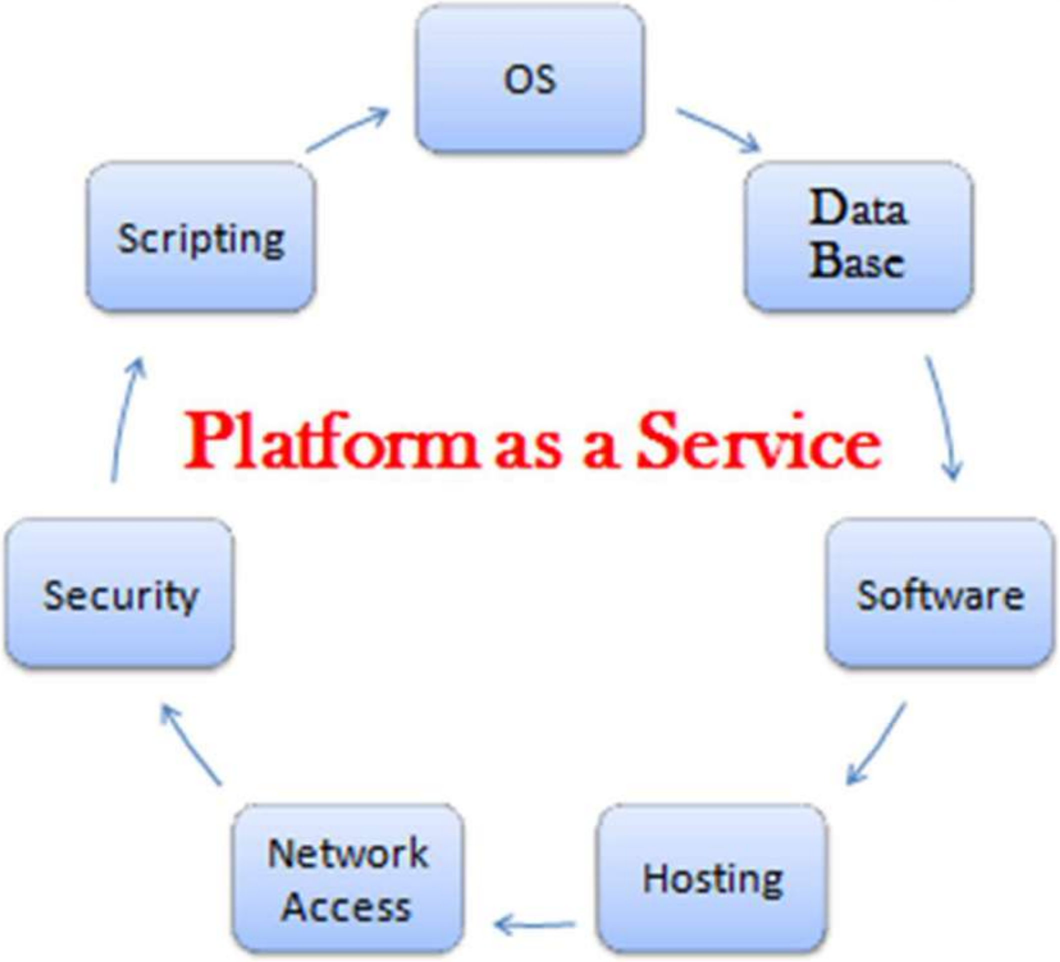
- an IoT development platform which is multipurpose middleware platform
- Built in end to end data encryption
- for monitoring, management, configuration of connected devices using communication protocols
- applications include smart homes, connected cars, fleet management…

MSRLT – Microsoft Research Lab of Things

- A platform for innovative solutions, applications and source code samples that have key features of enabling interconnection of devices
- Enables implementation, deployment, monitoring field studies and analysis of experimental data in healthcare, energy management, home automation

PaaS

- Connected Platform as a service Cloud
- A service model in which, a platform is made available to a developer of application, service or process on demand.
- these services are made available over internet – on demand.
- The platform, n/w, resources, maintenance, updating, security - as per developer requirement – and will be the responsibility of the service provider



Google Cloud Platform

Amazon Web Services

Microsoft Azure

Oracle Corporation

Cisco Systems

Carriots

- #1) Google Cloud **Platform**.
- #2) Particle.
- #3) Salesforce **IoT** Cloud.
- #4) ThingWorx.
- #5) IBM Watson **IoT**.
- #6) Amazon AWS **IoT** Core.
- #7) Microsoft Azure **IoT** Suite.
- #8) Samsung Artik Cloud.

Xively

- A PaaS model cloud platform
- Free for developers
- Permits many languages and platforms
- Scalable
- Manages and routes messages in realtime, has lifecycle management
- Does time series archieving, generates conditional triggers etc…
- Permits RESTful API and multiple data formats like XML, CSV etc…
- Also has several libraries for programming languages like Ruby, Python, Java etc…

IoT Applications Can be Divided Into the following categories

Premise Monitoring

- Internet connected camera, sensors etc… in banks, ATM's airport, shopping centres.
- **Eg: Internet Connected devices at banks, ATM's – premise monitoring**

Supply Chain Monitoring

- Internet connected RFID's, sensors, cameras etc… in production, distribution and services, supply chain order verification, automated reordering, shipping
- **Eg: SCOVARS – Supply Chain Order Verification, Automated Reordering and Shipping**

Customer Monitoring

- Internet connected digital devices, mobile apps, wearable devices - which can provide useful information about behaviors, preferences, health, locations of customers.
  Eg: TCCICDD – Tracking Of Customers Carrying Internet Connected Digital Devices

Product Monitoring

- Internet connected embedded hardware, software, devices and IIoT technologies which helps in product/production optimization.

Generalized Stages in Development of an IoT based application

Abstraction

- Ignoring the inessential details of things and dealing with the generalized interface of the model. It includes how to utilize data from a source. In short, define the purpose and manner in which, the system will operate

Architecture Reference Model

- Data flow diagrams and domain architecture reference models provides guidance on the system development.

Identifying Requirements of each domain

- Software and hardware requirements to be arranged at each layer/stage

Design Implementation

- Prototyping using necessary embedded platforms

Testing

- Check for errors in the system in a laboratory environment.

Development of Some Important IoT Applications

Connected ATM Premise Monitoring*

- An example of premise monitoring application.
- Implemented in banks, ATM's, offices, stores and other strategic locations for monitoring and security.,

SCOVARS – Supply Chain Order Verification, Automated Reordering and Shipping*

- Supply chain monitoring is important for companies, distributors and manufacturers.
- Includes planning and scheduling of production, scheduling deliveries, shipping, delivery confirmation, automated reordering, order verification, acknowledgement. (Cyclic operations)

TCCICDD – Tracking Of Customers Carrying Internet Connected Digital Devices*

- Data from tracking of customers and their database provides behaviours, preference, locations, usage patterns and product health.
- Applications like business planning, analytics, health, services and manufacturing use this data.
- Tracking is done using customer's internet connected mobile apps and wearable devices, customer databases and customer end embedded devices.

***The various development stages (including architectural reference model) of the above examples may be learnt from material provided.**

Some Important IoT Application Developments

Smart Homes*

- Implemented using open source softwares like OpenHAB
- Home lighting control, monitoring appliances, security, intrusion detection, video surveillance, access control, security alerts, WiFi control and controlling
- Uses smart lighting, proximity sensors, intrusion sensors, appliance control interfaces

*The various development stages (including architectural reference model) of the above examples may be learnt form material provided.

Smart City*

- Defined as a vision which integrates multiple ICT (Information and Communication technologies) and IoT solutions in a secure fashion to manage a city's assets like Information systems, schools, libraries, transportation, hospitals, power plants, water supply systems, waste management, law enforcement etc…
  - 1. Smart parking spaces
  - 2. Smart street lightings and smart lighting solutions
  - 3. Smart traffic solutions, smart energy management, smart parking, smart waste bins, smart street lighting, and security and surveillance
  - 4. Smart water management for monitoring and optimizing a city's water and sewage services
  - 5. Smart connected bike share services
  - 6. Smart health services
  - 7. Smart structures (building, bridges and historical monuments) health, vibrations and material conditions monitoring, analysing and managing structures health data to improve energy usage, maintenance, operations, and comfort solution.

***The various development stages (including architectural reference model) of the above examples may be learnt form material provided.**

Smart City Parking

- 1. Guides the drivers for the available parking slots and spaces
- 2. Provides a mobile app, and the app assists a driver and enables him/her to obtain the appropriate parking-slot information remotely. The information includes location of the parking utility, its cost, advance reservation facility, direction guidance and the time to reach an available slot. The app accesses the slots availability in real-time.
- 3. Publishes messages in real time for available slots and alerts for slot unavailability at the parking utility
- 4. Consists of a central supervisory control and monitoring system (CSS) which connects the edge sensors and devices, accurately senses the slots available for occupancy
- of vehicles in real time, and predicts the expected availability time in case of nonavailability of slots
- 5. Optimises the usages of parking spaces and reaching time
- 6. Provides display boards at road traffic junctions for status of availability
- 7. Provides good parking experience to users
- 8. Adds value for all parking stakeholders, drivers and service providers
- 9. Enables intelligent decisions using data and historical analytics reports at city cloud data store, and enables planning for traffic flow in the city by predictive analytics

Smart Environment-Monitoring

- 1. Preparations for assessment of environment impact
- 2. Establish the trends in environmental parameters and current status of the environment
- 3. Interpretation of data and evaluate environmental quality indices
- 4. Monitor the air, soil and water quality parameters
- 5. Monitor harmful chemicals, biological, microbiological, radiological and other parameters

Weather Monitoring System

- Each measuring node for weather parameters is assigned an ID. Each lamppost deploys a wireless sensor node. Each node measure the T, RH and other weather parameters at assigned locations. A group of WSNs communicates using ZigBee and forms a network. Each network has an access point, which receives the messages from each node. Each access point associates a gateway. Forward and store the parameters on an Internet cloud platform

  - 1. Publishes weather messages for the display boards at specific locations in the city and communicates to weather API at mobile and web users
  - 2. Publishes the messages in real time and send alerts using a weather reporting application
  - 3. Analyse and assess the environment impact
  - 4. Enables intelligent decisions using data and historical analytics reports at city cloud weather data store

Air Pollution Monitoring

- 1. Monitoring and measuring levels of CO, a gas dangerous above 50–100 ppm level; carbon dioxide (CO2), a gas causes which greenhouse effect; and ozone (O3), a gas dangerous above 0.1 mg/per kg air level, for controlling air pollution
- 2. Monitoring and measuring levels of hydrogen sulfide (H2S), a highly toxic gas. It is a greenhouse gas so its increase may contribute to global warming as well.
- 3. Monitoring and measuring levels of hydrocarbons, such as ethanol, propane.
- 4. Measure T, RH and Patm parameters for calibrations of sensed gaseous parameters of each node
- 5. Investigate air quality and the effects of air pollution.
- 6. Compute Air Quality Index (AQI) from the parameters, such as hourly or daily
- averages of air pollutant concentration, particulate matter (such as dust or carbon
- particle)
- 7. Compute source and spatial dispersion of pollutants as a function of day conditions,
- wind-speed and direction, air temperature and air temperature gradient with altitude
- and topography using analytics.
- 8. Data visualisation
- 9. Report the pollution status to monitoring authorities

Smart Irrigation

- Sensors for moisture and actuators for watering channels are used in smart irrigation.
- Uses soil moisture sensors with a sensor circuitry board with each one installed at certain depth in the fields.
- Uses an array of actuators (solenoid valves) which are placed along the water channels and that control deficiencies in moisture levels above thresholds during a given crop period.
- Uses sensors placed at three depths for monitoring of moisture in fruit plants such as grapes or mango, and monitors evaporation and transpiration
- Measures and monitors actual absorption and irrigation water needs
- Each sensor board is in a waterproof cover and communicates to an access point using ZigBee protocol. An array of sensor circuits forms a WSN.

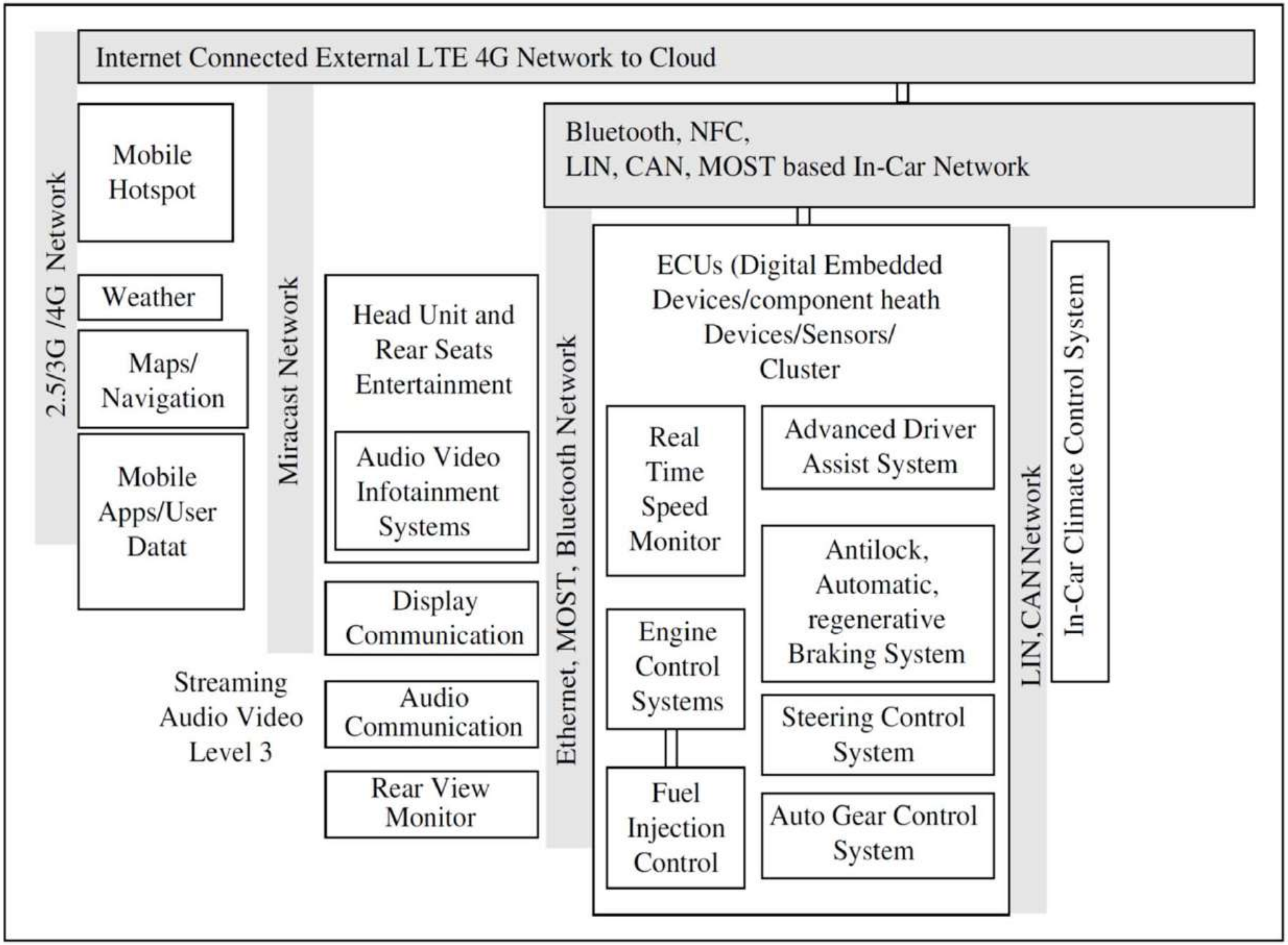
CONNECTED CARS

Connected Cars

- Internet connection enables automotive service centers and helps in predictive maintenance
- Data is generated using Electronic control Units inside the car
- Bluetooth, NFC, Controller Area Network, LAN, MOST (Media Oriented System Transport)

Cluster consists of

- Engine control
- Speed control
- Brake – ABS, automatic braking, regenerative braking, EBD…
- Safety – airbags, auto-hill assist…
- Ergonomics
- Car environment controls
- Automobile status monitoring


Internet Connected External LTE 4G Network to Cloud
2.5/3G /4G Network
Mobile Hotspot
Weather
Maps/ Navigation
Mobile Apps/User Datat
Miracast Network
Head Unit and Rear Seats Entertainment
Audio Video Infotainment Systems
Display Communication
Audio Communication
Rear View Monitor
Streaming Audio Video Level 3
Bluetooth, NFC,
LIN, CAN, MOST based In-Car Network
Ethernet, MOST, Bluetooth Network
ECUs (Digital Embedded Devices/component heath Devices/Sensors/ Cluster
Real Time Speed Monitor
Engine Control Systems
Fuel Injection Control
Advanced Driver Assist System
Antilock, Automatic, regenerative Braking System
Steering Control System
Auto Gear Control System
LIN,CANNetwork
In-Car Climate Control System

Overview of internet connected car

Routes & Traffic monitors

- Using mobile API based car location ,maps, traffic reports, route planners

Infotainment

- Speech to machine conversion
- Touch panels etc…

Internet connectivity

- 4G or WiFi networks
- Enables weather, maps, navigation etc…
- Location API's – use real time GPS location
- Internet connectivity also from nearby hotspots installed
- Live video/audio streaming

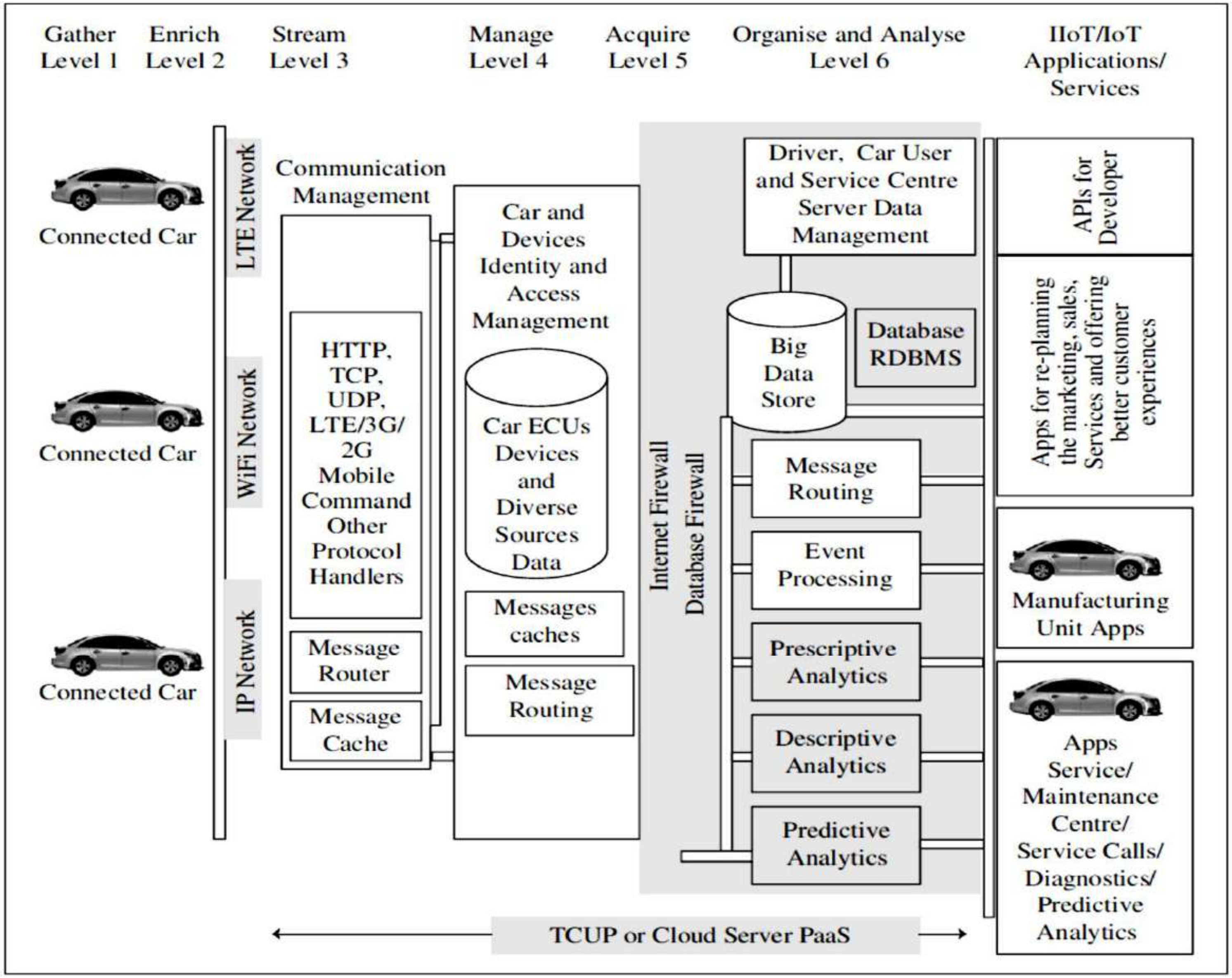
Applications of connected cars

1. Automotive components predictive automotive maintenance service (ACPAMS)

- Optimising service – when and where required
- Automatic detection of service requirement
- Direct transfer of info to service centres
- Driver reminding service

2. Re-planning manufacturing process (RPMP)

- Data from car is used for improving manufacturing process
- Design better customer experience


Gather
Level 1
Enrich
Level 2
Stream
Level 3
Manage
Level 4
Acquire
Level 5
Organise and Analyse
Level 6
IIoT/IoT
Applications/
Services
Connected Car
Connected Car
Connected Car
LTE Network
WiFi Network
IP Network
Communication
Management
HTTP,
TCP,
UDP,
LTE/3G/
2G
Mobile
Command
Other
Protocol
Handlers
Message
Router
Message
Cache
Car and
Devices
Identity and
Access
Management
Car ECUs
Devices
and
Diverse
Sources
Data
Messages
caches
Message
Routing
Internet Firewall
Database Firewall
Driver, Car User
and Service Centre
Server Data
Management
Big
Data
Store
Database
RDBMS
Message
Routing
Event
Processing
Prescriptive
Analytics
Descriptive
Analytics
Predictive
Analytics
APIs for
Developer
Apps for re-planning
the marketing, sales,
Services and offering
better customer
experiences
Manufacturing
Unit Apps
Apps
Service/
Maintenance
Centre/
Service Calls/
Diagnostics/
Predictive
Analytics
TCUP or Cloud Server PaaS

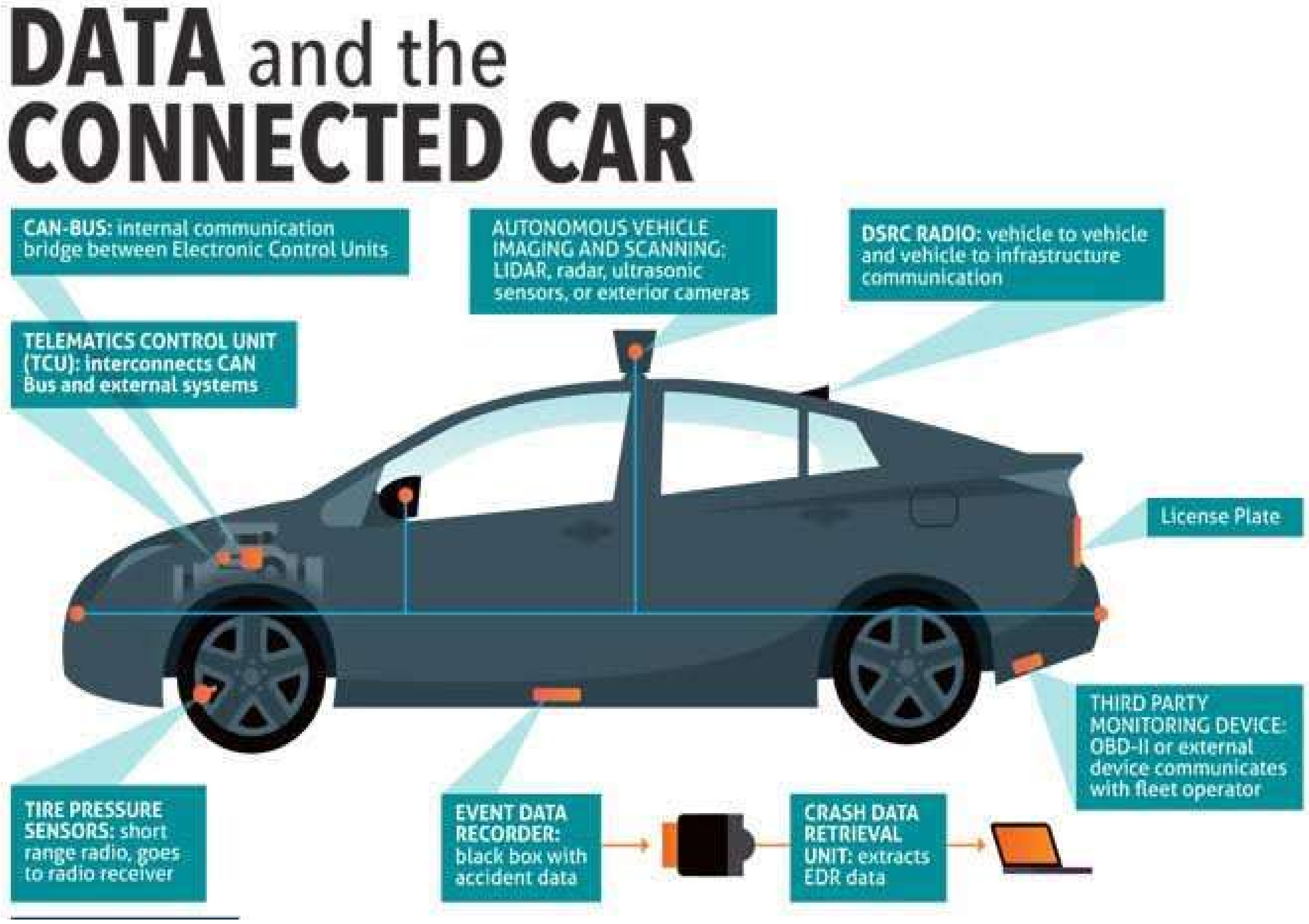
Layer 1 & 2

- Using mobile Apps
- Embedded sensors
- Device softwares
- Design of ECU
- May use Raspberry Pi / Arduino / intel boards
- Eclipse IoT

Layer 3,4,5,6

- Telematics control unit and Any server platform PaaS (Xively)
- Communication management
- Data store
- End applications

69°
69°



DATA and the CONNECTED CAR

CAN-BUS: internal communication bridge between Electronic Control Units

AUTONOMOUS VEHICLE IMAGING AND SCANNING: LIDAR, radar, ultrasonic sensors, or exterior cameras

DSRC RADIO: vehicle to vehicle and vehicle to infrastructure communication

TELEMATICS CONTROL UNIT (TCU): interconnects CAN Bus and external systems

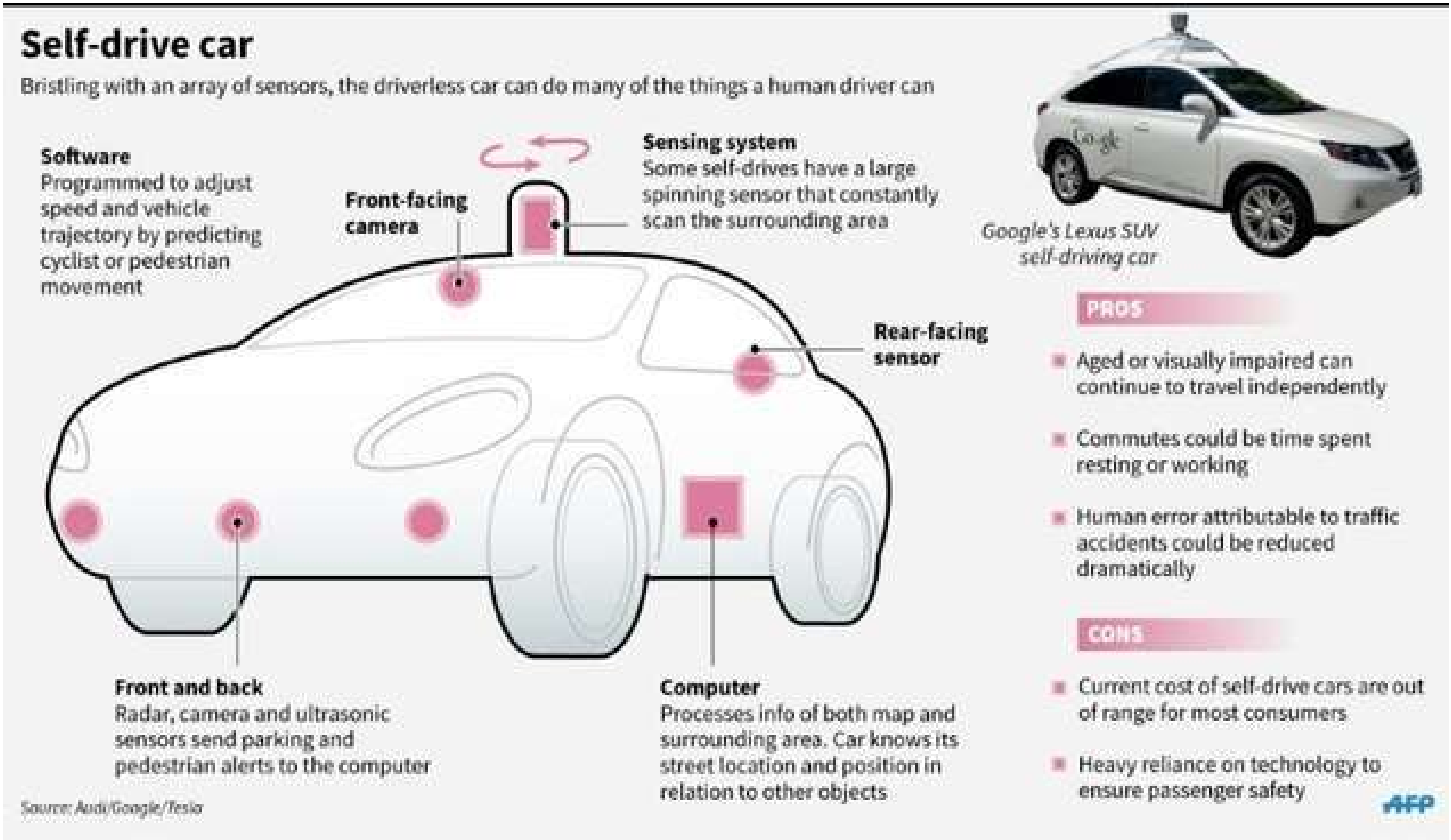
License Plate

THIRD PARTY MONITORING DEVICE: OBD-II or external device communicates with fleet operator

TIRE PRESSURE SENSORS: short range radio, goes to radio receiver

EVENT DATA RECORDER: black box with accident data

CRASH DATA RETRIEVAL UNIT: extracts EDR data



Self-drive car

Bristling with an array of sensors, the driverless car can do many of the things a human driver can

Software
Programmed to adjust speed and vehicle trajectory by predicting cyclist or pedestrian movement

Front-facing camera

Sensing system
Some self-drives have a large spinning sensor that constantly scan the surrounding area

Rear-facing sensor

Front and back
Radar, camera and ultrasonic sensors send parking and pedestrian alerts to the computer

Computer
Processes info of both map and surrounding area. Car knows its street location and position in relation to other objects

Google's Lexus SUV self-driving car

PROS

- Aged or visually impaired can continue to travel independently
- Commutes could be time spent resting or working
- Human error attributable to traffic accidents could be reduced dramatically

CONS

- Current cost of self-drive cars are out of range for most consumers
- Heavy reliance on technology to ensure passenger safety

Source: Audi/Google/Tesla

AFP

Connected Vehicles – Building a 360° View

What's the average speed?

What kind of fuel is being used?

How is the engine performing?

When is the next maintenance required?

What are their in app purchases?

What features are being used the most/least?

When is the best time to drive to the office?

Where did they go on their last trip?

Who do they know that might buy our vehicles?

cloudera

AUTOPILOT: Musk promises Tesla owners a HANDS-OFF hands-on

Miracle driverless car software upgrade

In October 2014, Tesla Motors unveiled the world's first dual electric motor production car and announced that new safety and autopilot hardware is standard on every new Model S.

Automatic emergency braking, which will attempt to automatically halt the car when it detects that you're about to slam into something else. The Model S' self-steering will also allow you to effectively "summon" the vehicle. Push a button, and it'll leave wherever you've parked it and find its way to you.