Internet of Things

Introduction

Ask Google.. ☺

Where is my key?

Where are my kids?

What are the “Things” in the IoT?

- Could be anything
  - **Physical**
  - **Virtual**

According to Wikipedia,
IoT refers to the interconnection of uniquely identifiable embedded computing-like devices within the existing Internet infrastructure.

In simple words

Internet of Things: Anetwork of internet-connected objects able to collect and exchange data using embedded sensors.

Physical

- *Objects such as*
  - Climate control
  - Security/Disaster alarm system
  - Energy/Resource metering (Electricity, Gas, Water)
  - Water boiler, Solar boiler
  - Car (OBDII, vehiclepi)
  - The Sun (sunrise/set times, azimuth)
- *Living things*
  - People (self quantification)
    - Presence
    - Location
    - Health metrics
      - Weight
      - Activity tracking
      - Temperature
      - Blood Glycose monitors
      - Fitbit
  - Pets

Virtual

- *Personal schedule/calendar*
- *Social*
  - Email
  - Twitter/FB
  - Online notification platforms
- *WWW resources*
  - Weather forecasting
  - Stocks
  - Traffic
  - DB Storage
  - Visualization dashboards

What is IoT?

- The Internet of Things (IoT) is the network of physical objects or "things" embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data.

- IoT allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration between the physical world and computer-based systems, and resulting in improved efficiency, accuracy and economic benefit.

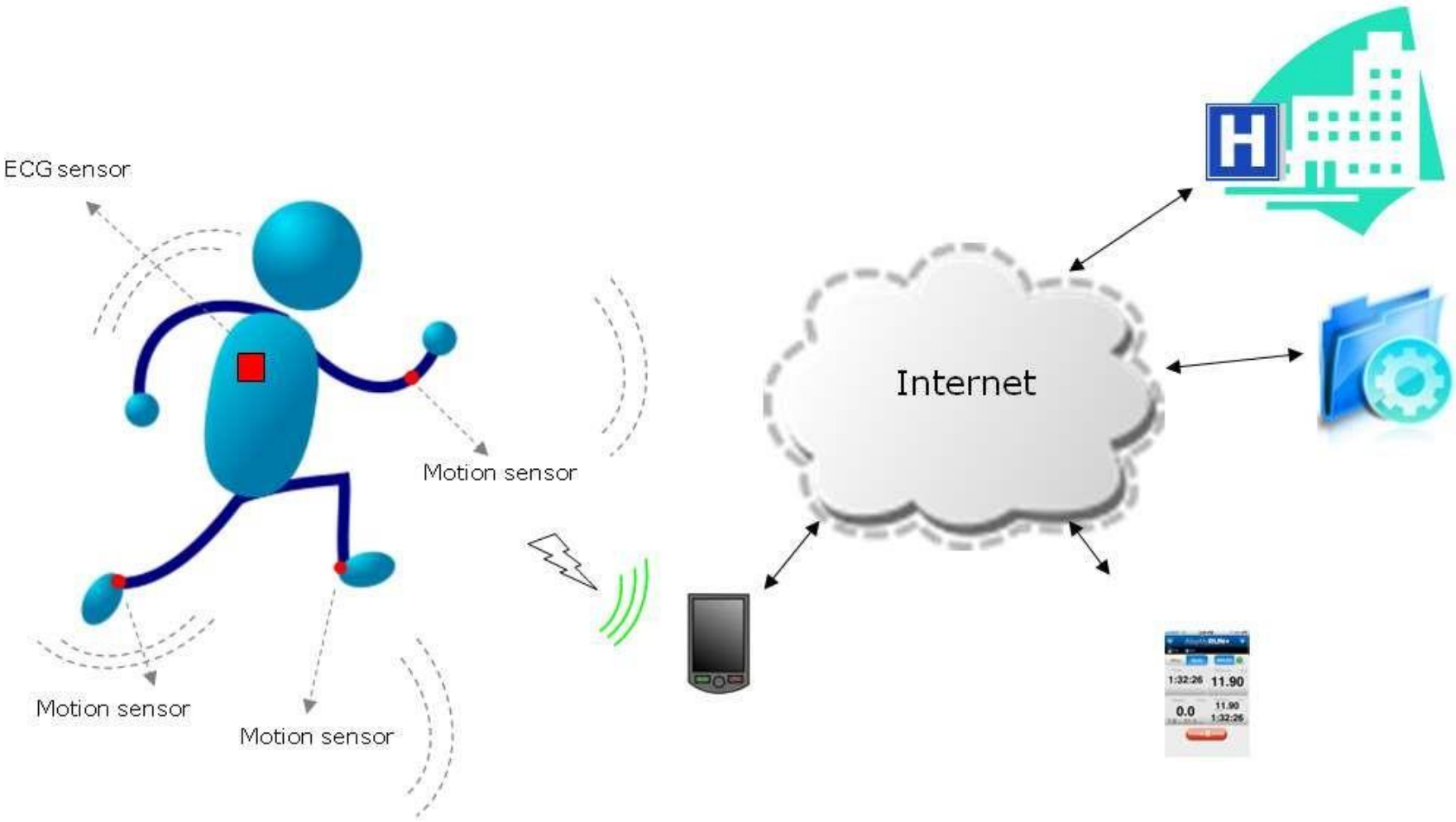
Let's start with understanding a few things

- ✓**Broadband Internet is becoming more widely available**
- ✓**The cost of connecting is decreasing**
- ✓**More devices are being created with Wi-Fi capabilities and sensors built into them**
- ✓**Technology costs are going down and smartphone penetration is sky-rocketing**

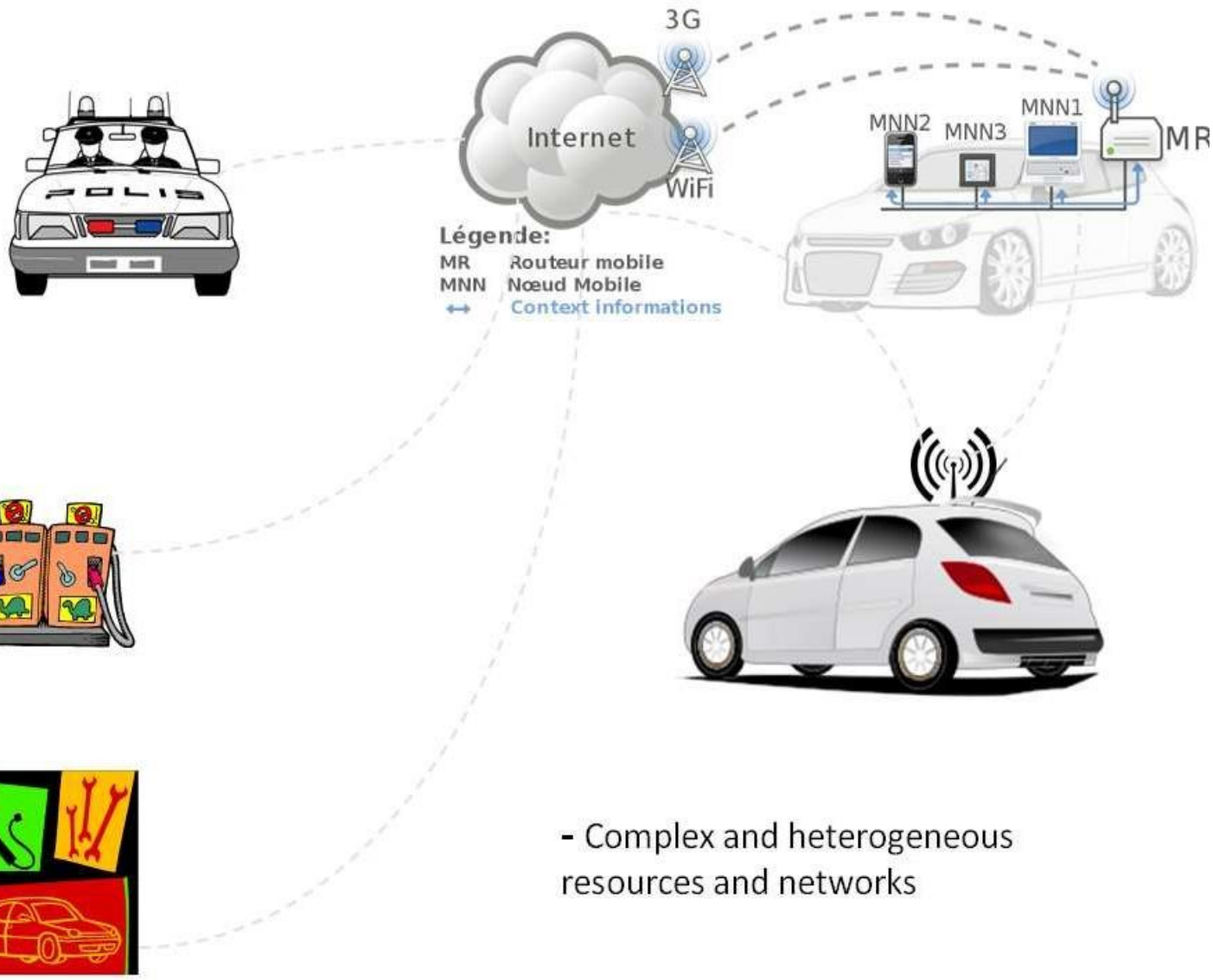
All of these things are creating a "perfect storm" for the IoT.

- This is the concept of basically connecting **any device** with an on and off switch to the Internet.
- This includes everything from ***cellphones, coffee makers, washing machines, headphones, lamps, wearable devices*** and almost anything else you can think of.

IOT: People connecting with Things

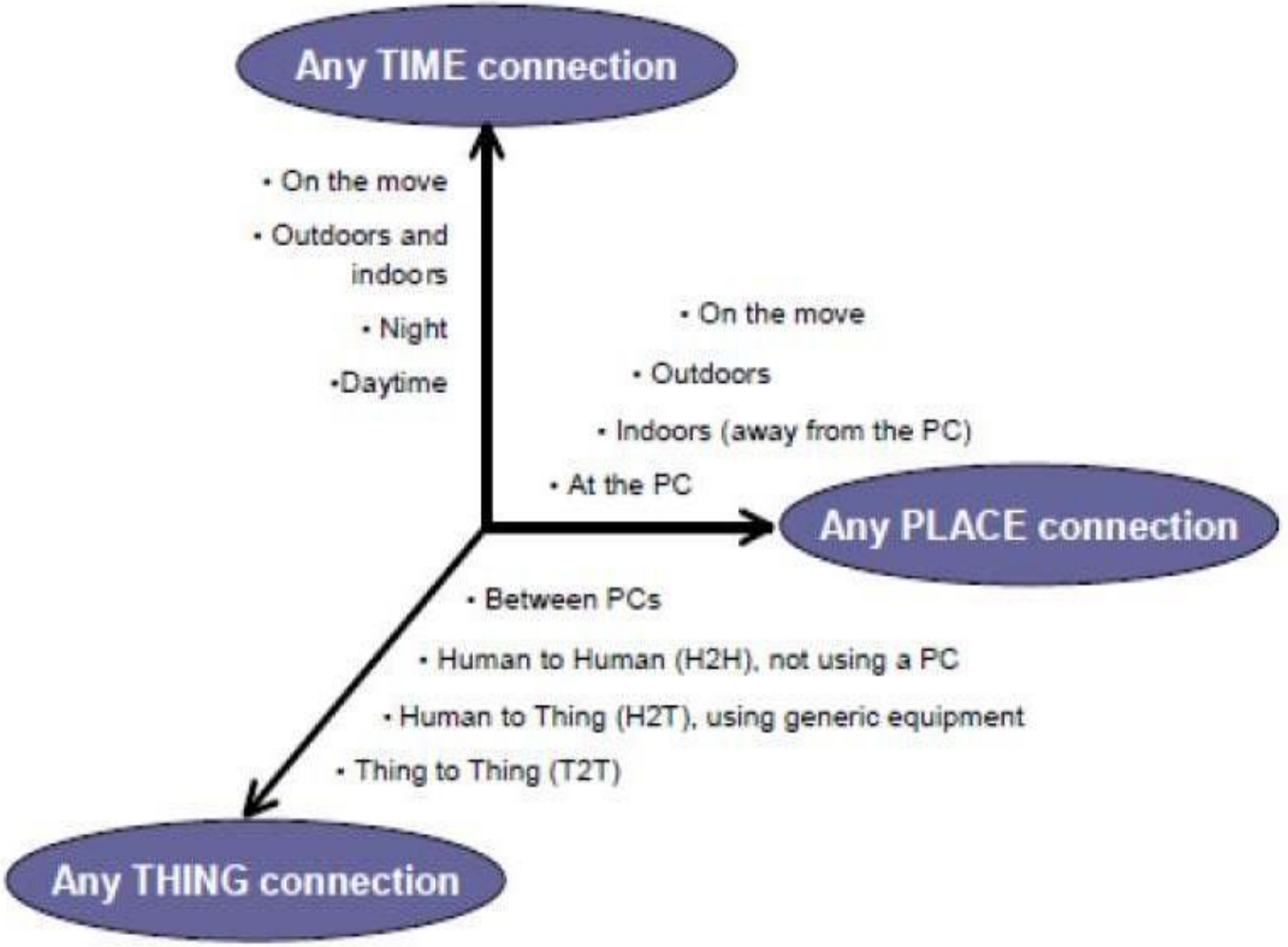


IoT: Things connecting with Things



- Complex and heterogeneous resources and networks

We will now have connectivity for anything. From any time, any place connectivity for anyone



How Does IOT Impact You?

The new rule for the future is going to be**, "Anything that can be connected, will be connected."**

But why on earth would you want so many connected devices talking to each other?

There are many examples for what this might look like or what the potential value might be.

Say for example

✓You are on your way to a meeting; your car could have access to your calendar and already know the best route to take. If the traffic is heavy your car might send a text to the other party notifying them that you will be late.

✓What if your alarm clock wakes up you at 6 a.m. and then notifies your coffee maker to start brewing coffee for you?

✓What if your office equipment knew when it was running low on supplies and automatically re-ordered more?

Various Names, One Concept

For over a decade after the introduction of the term Internet-of-Things, different organizations and working groups have been providing various definitions.

- M2M (Machine to Machine)
- "Internet of Everything" (Cisco Systems)
- "World Size Web" (Bruce Schneier)
- "Skynet" (Terminator movie)
- Cloud of Things
- Web of Things

➢ All these terms are very relevant (and in most cases overlapping) to IoT. Nevertheless, they have also have differences from IoT.

➢ In gerenal there are different viewpoints for IoT, and IoT experts approach IoT from different angles.

For example:

❑ The **"Things-Oriented"** viewpoint focuses on technologies for the representation and use of the things e.g., RFID (Radion-Frequency Identification), NFC (Near Field Communications), WSN (Wireless Sensor Networks), Things connectivity technologies etc.

❑ The **"Internet-Oriented"** viewpoint focuses on the internet and web aspects of IoT, such as the web-of-things layer for simplifying application development, IPv6 for internet connectivity and identification etc.

Independently of one's viewpoint about IoT and IoT technologies, any non-trivial IoT system is expected to comprise the following elements:

•Sensor: a device that converts a physical parameter to an electrical output.
Sensors: light, temperature, humidity, sound, distance, movement, position, orientation, acceleration ect.

• Actuator: a device that converts an electrical signal to a physical output.
Actuators: display things, make sound or light, vibrate, rotate, and translate ect.

- ✓Sensors and Actuators.
- ✓Communication infrastructure between servers or server platforms.
- ✓Server/Middleware Platforms.
- ✓Data Analytics Engines.
- ✓Apps (iOS, Android, Web).

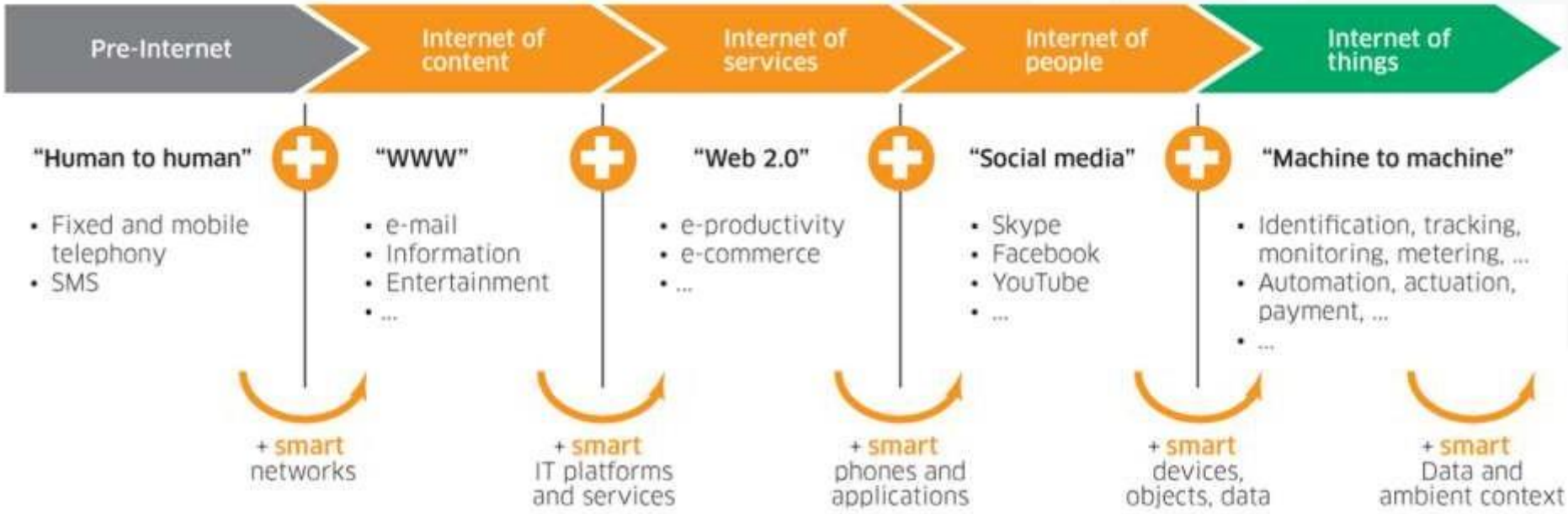
computer software that provides services to software applications beyond those ... *Middleware* includes web *servers*, application *servers*, content management systems, and similar tools that support

Transforming Data to Decisions

History/ Evolution of IOT

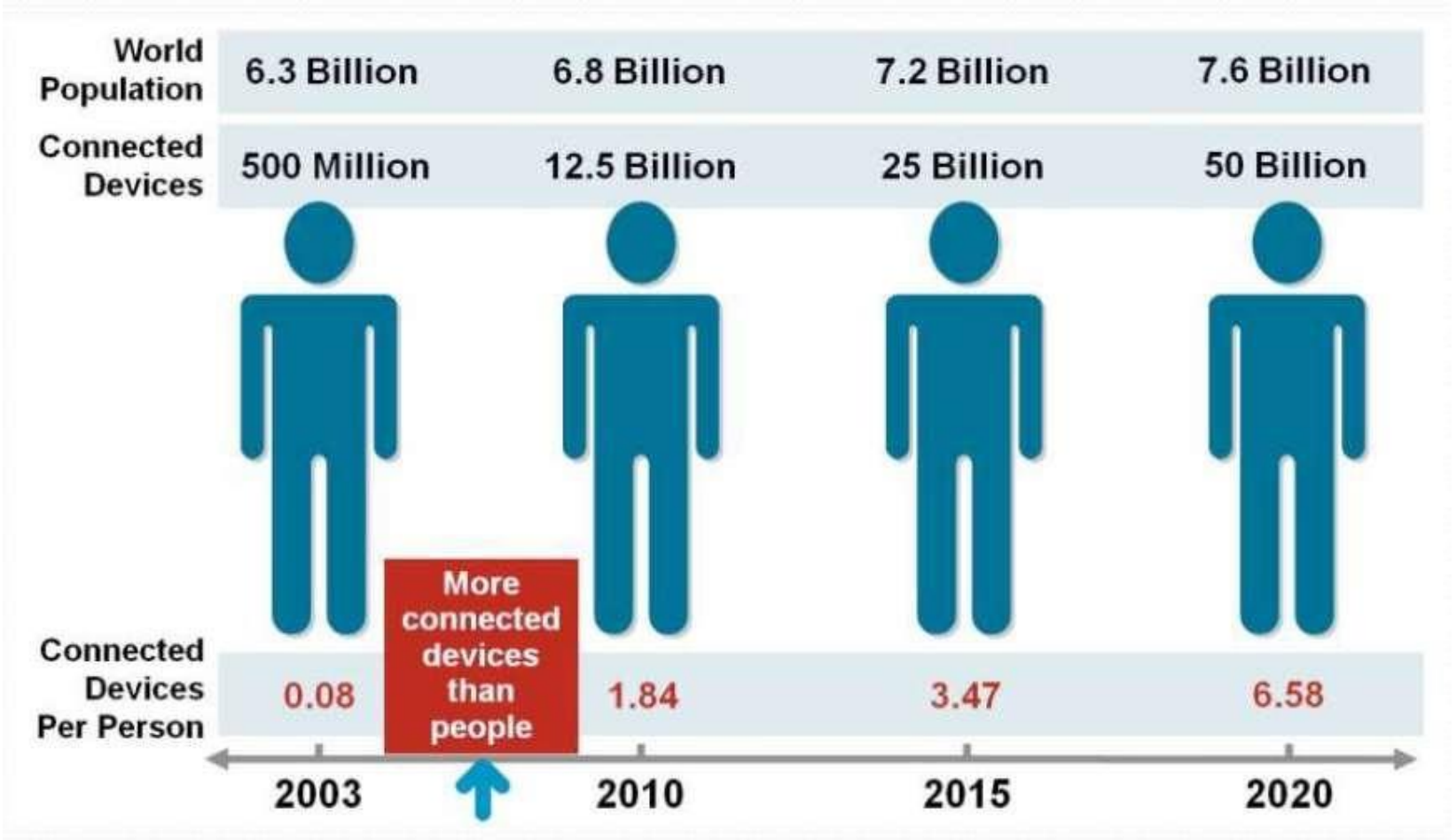
- The concept of the Internet of Things first became popular in **1999**, through the Auto-ID Center at **MIT**
- The term IoT was first coined by **Kevin Ashton** in 1999.
- Radio-frequency identification (**RFID**) was seen as a prerequisite for the IoT at that point.
- If all objects and people in daily life were equipped with identifiers, computers **could manage and inventory them**.
- Besides using RFID, the tagging of things may be achieved through such technologies as near field communication(**NFC**), **barcodes**, **QR** codes, **bluetooth** ect.

✓**Major industrial initiative is now being undertaken by large corporations ,where the Machine to machine (M2M) name is more commonly used**


Pre-Internet
Internet of content
Internet of services
Internet of people
Internet of things
"Human to human"
• Fixed and mobile telephony
• SMS
"WWW"
• e-mail
• Information
• Entertainment
• ...
"Web 2.0"
• e-productivity
• e-commerce
• ...
"Social media"
• Skype
• Facebook
• YouTube
• ...
"Machine to machine"
• Identification, tracking, monitoring, metering, ...
• Automation, actuation, payment, ...
• ...
+ smart networks
+ smart IT platforms and services
+ smart phones and applications
+ smart devices, objects, data
+ smart Data and ambient context

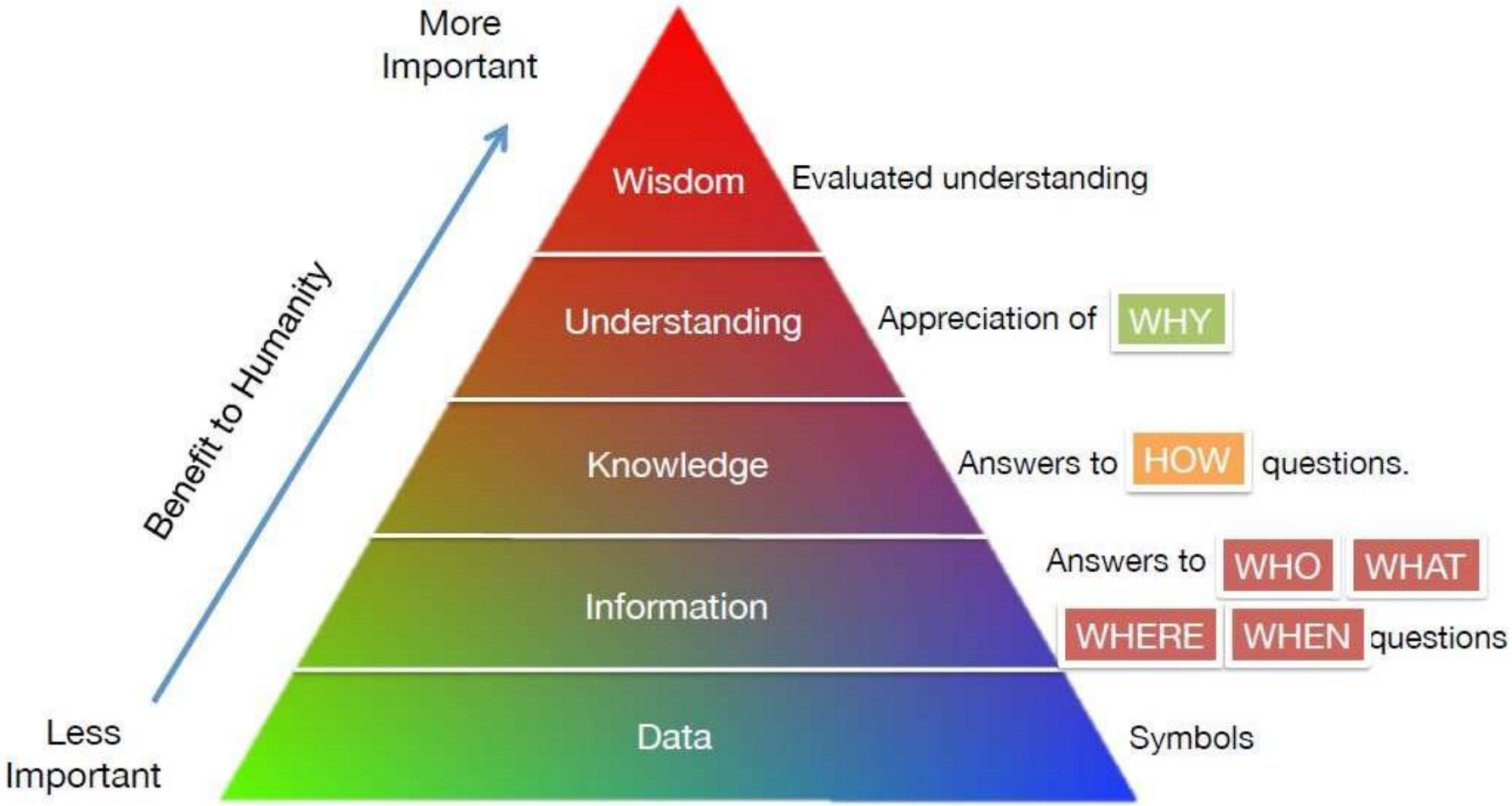
Current Status & Future Prospect of IoT

"THINGS" vs "People"



"Change is the only thing permanent in this world"

Data and Information

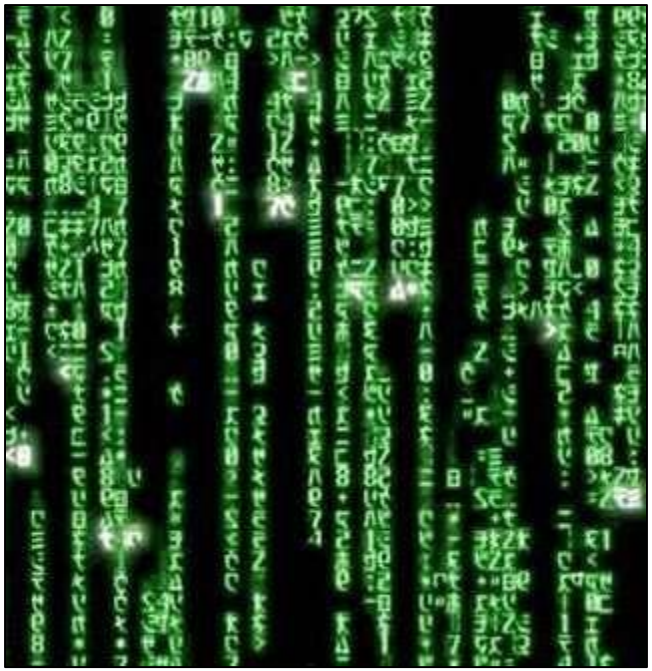


The more data that is created, the better understanding and wisdom people can obtain.

Data

- Data **are** raw facts and figures that on their own have no meaning

- These can be any alphanumeric characters i.e. **text, numbers, symbols ect**



Data Examples

- Yes, Yes, No, Yes, No, Yes, No, Yes
- 42, 63, 96, 74, 56, 86
- 111192, 111234

- None of the above data sets have any meaning until they are given a **CONTEXT** and **PROCESSED** into a useable form

Data Into Information

- To achieve its aims the organisation will need to **process** data into information.
- Data needs to be turned into meaningful information and presented in its most useful format
- Data must be processed in a **context** in order to give it meaning

Information

- Data that has been processed within a context to give it meaning

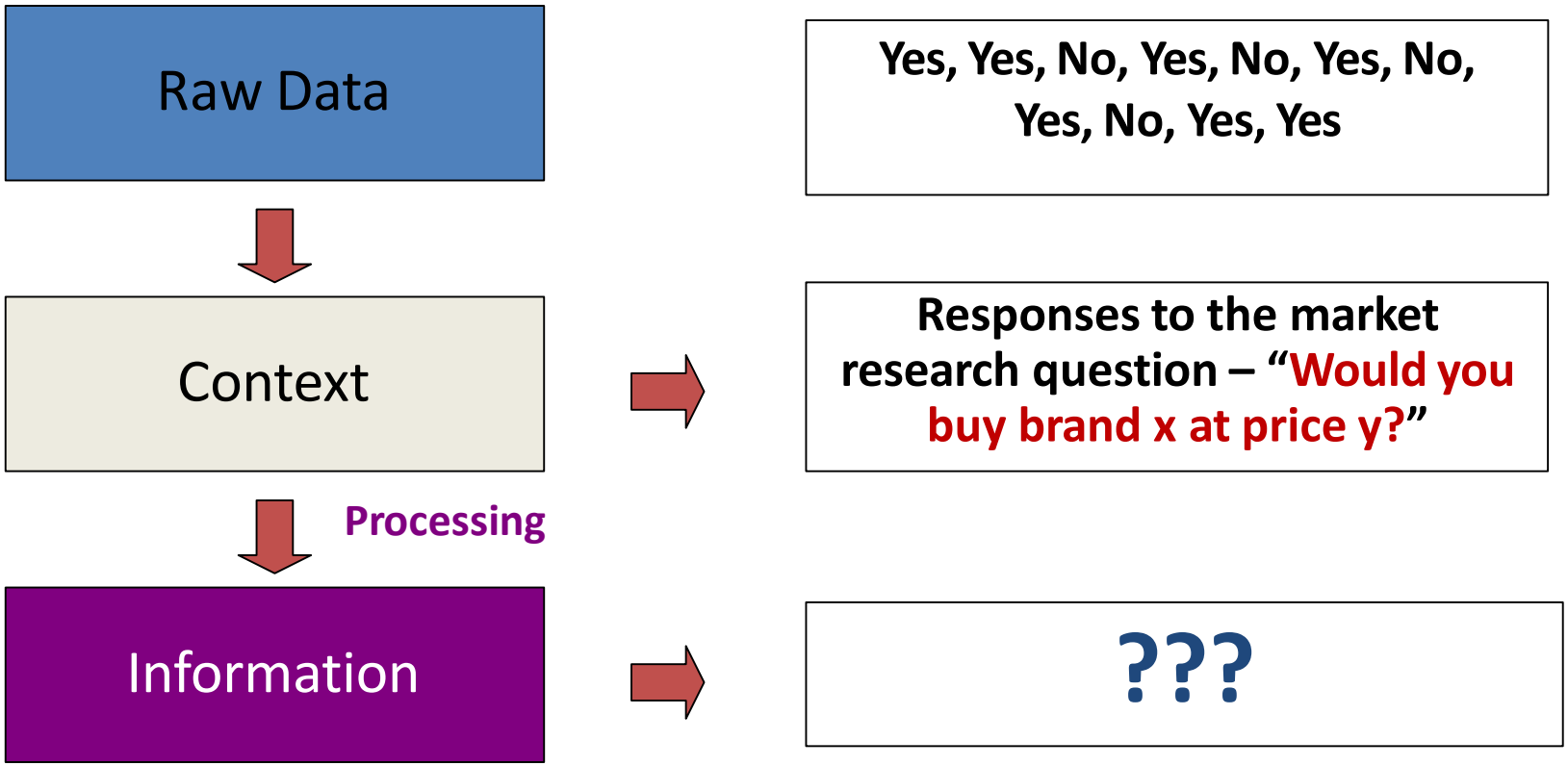
OR

- Data that has been processed into a form that gives it meaning

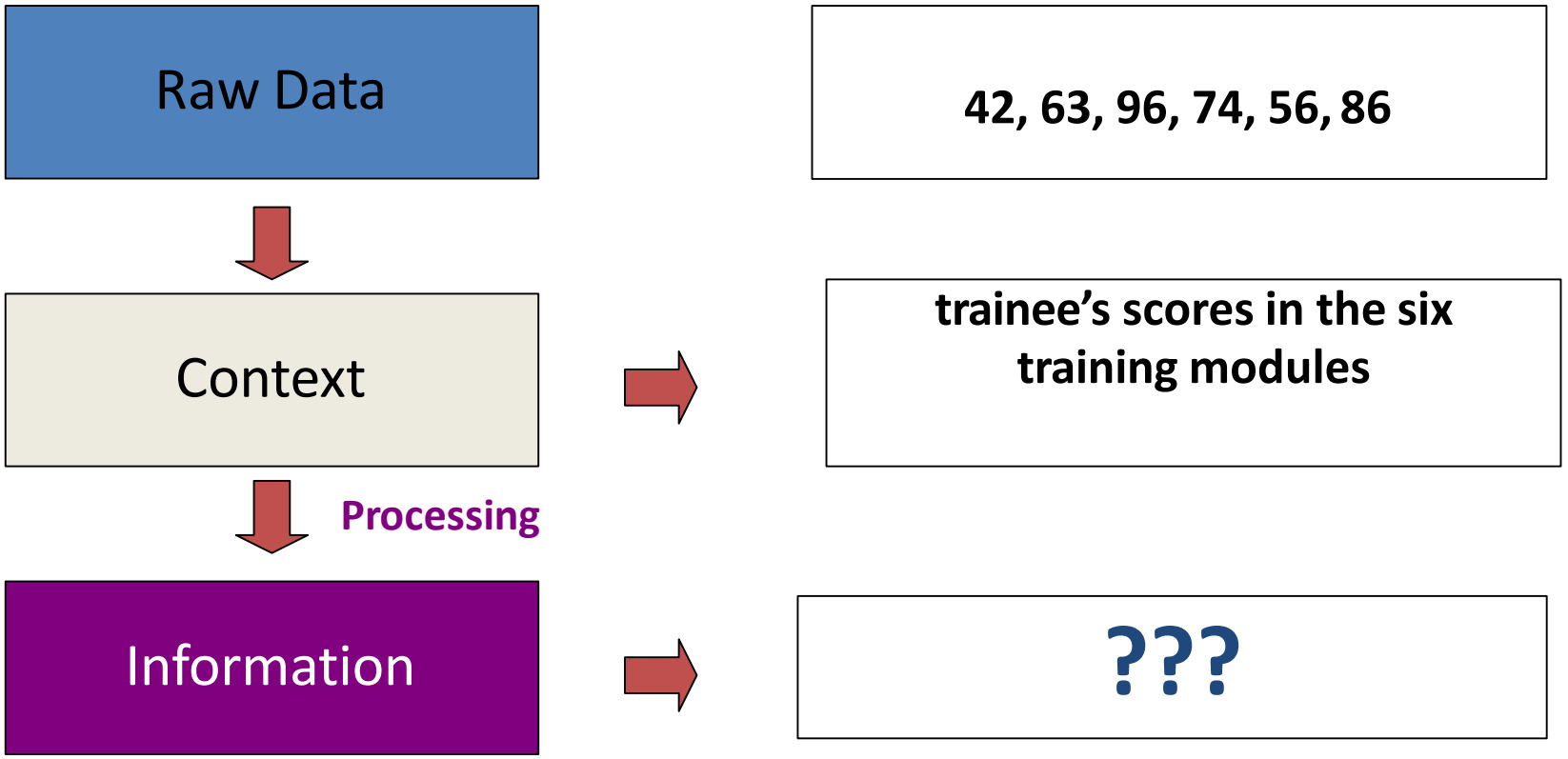
Examples

- The next 3 examples explain how the data could be processed to give it meaning
- What information can then be derived from the data?

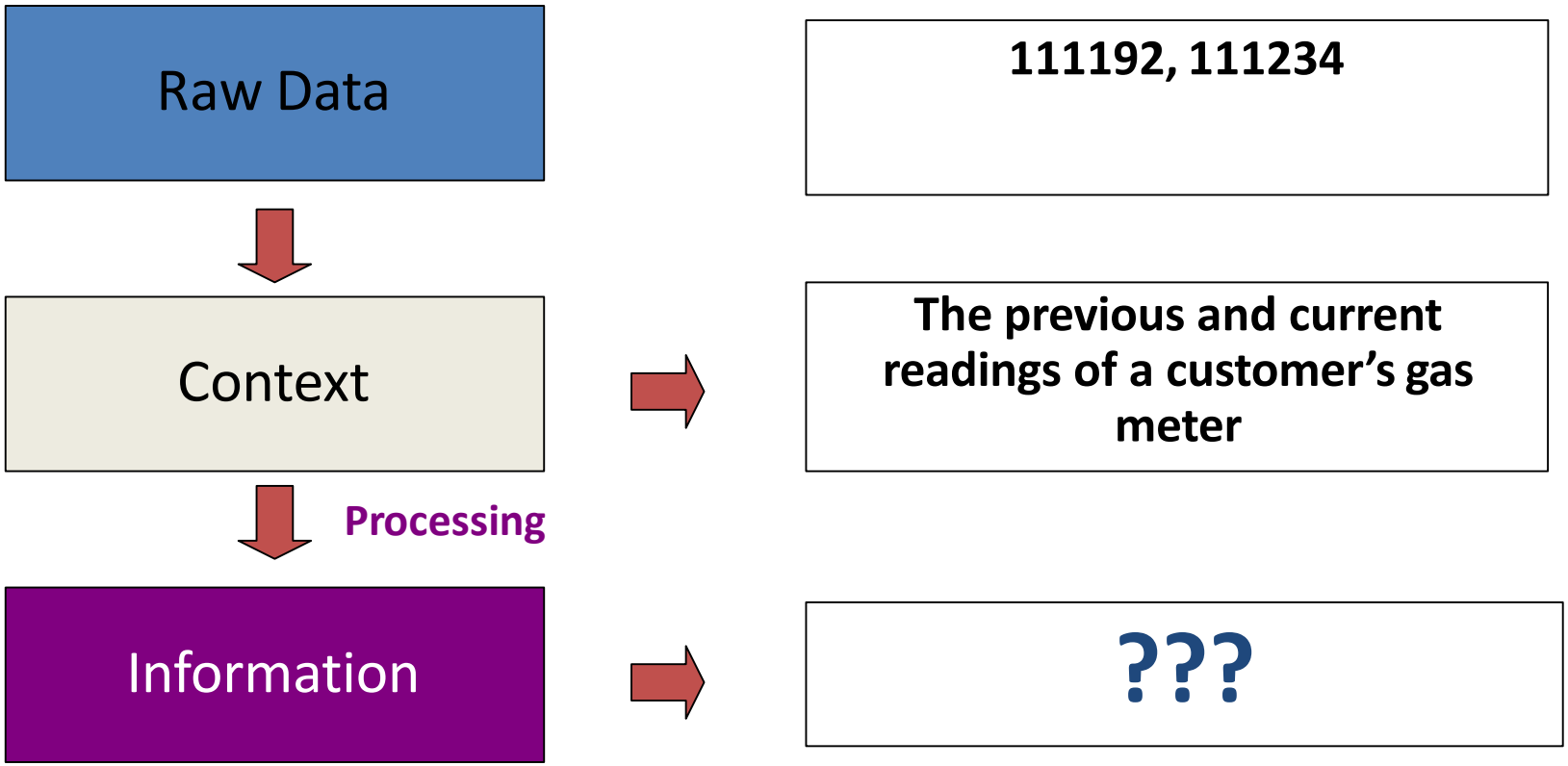
Example 1


Raw Data
Context
Processing
Information
Yes, Yes, No, Yes, No, Yes, No, Yes, No, Yes, Yes
Responses to the market research question – "Would you buy brand x at price y?"
???

Example 2


Raw Data
42, 63, 96, 74, 56, 86
Context
trainee's scores in the six training modules
Processing
Information
???

Example 3


Raw Data
Context
Processing
Information
111192, 111234
The previous and current readings of a customer's gas meter
???

Knowledge

- Knowledge is the understanding of rules needed to interpret information

"…the capability of understanding the relationship between pieces of information and what to actually do with the information"

-Debbie Jones

Knowledge Examples

- **Using the 3 previous examples:**
  - **A Marketing Manager could use this information to decide whether or not to raise or lower price y**
  - **Trainee's teacher could analyse the results to determine whether it would be worth him re-sitting a module**
  - **Looking at the pattern of the customer's previous gas bills may identify that the figure is abnormally low and they are fiddling the gas meter!!!**

From IOT point of view

Data

Raw and unprocessed data obtained from IOT devices

Information

Inferred/summarized from data by filtering, processing, categorizing, condensing and contextualizing data

Knowledge

Inferred from information by structuring/organizing information and is put into action to achieve specific objectives.

Example: Weather Monitoring System

This doesn't have any meaning

Consider a series of raw sensor measurements ((72,45);(84,56)) generated by a weather monitoring system.

To give meaning, context is added ← Data represents the **temperature** and **humidity** measured every minute

With this context added we know the meaning of measured data tuples.

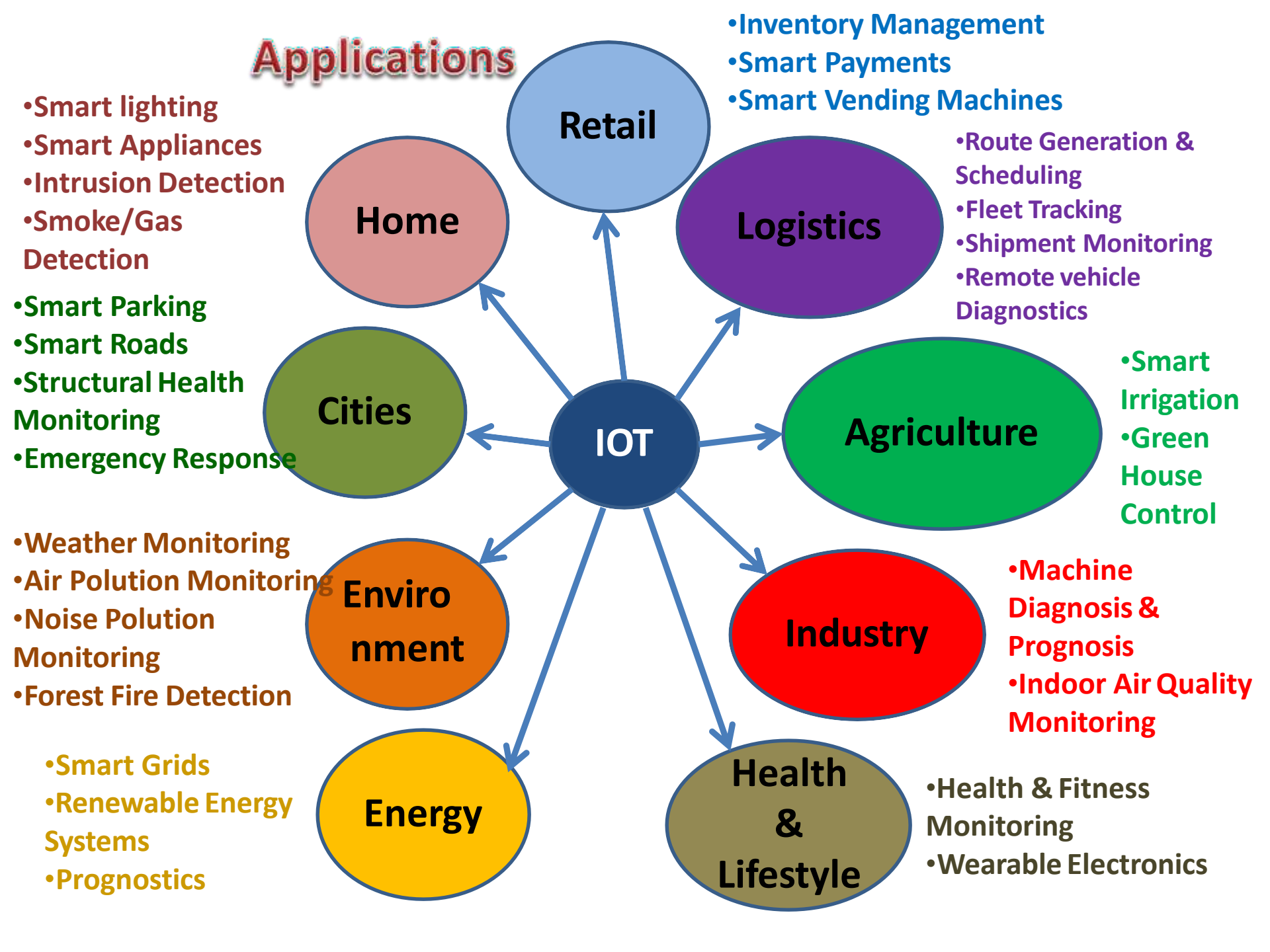
Information is obtained by categorizing, condensing or processing this data.

Example: Average temperature and humidity readings for last 5 minutes is obtained by averaging the last data tuples.

Organize the information and understand the relationship between pieces of information to infer **knowledge** which can be put into **action**.

*Example: An **alert** is raised if average temp in last 5 minutes exceeds 120F*

*And **alert** is shown on user's graphical position as well.*


Applications
IOT
Retail
•Inventory Management
•Smart Payments
•Smart Vending Machines
Home
•Smart lighting
•Smart Appliances
•Intrusion Detection
•Smoke/Gas Detection
Logistics
•Route Generation & Scheduling
•Fleet Tracking
•Shipment Monitoring
•Remote vehicle Diagnostics
Cities
•Smart Parking
•Smart Roads
•Structural Health Monitoring
•Emergency Response
Agriculture
•Smart Irrigation
•Green House Control
Enviro nment
•Weather Monitoring
•Air Polution Monitoring
•Noise Polution Monitoring
•Forest Fire Detection
Industry
•Machine Diagnosis & Prognosis
•Indoor Air Quality Monitoring
Energy
•Smart Grids
•Renewable Energy Systems
•Prognostics
Health & Lifestyle
•Health & Fitness Monitoring
•Wearable Electronics

1.1.1 Definition & Characteristics of IOT

Definition

A **dynamic global network** infrastructure with **self-configuring** capabilities based on standard and **interoperable communication protocols** where physical and virtual "things" have **identities, physical attributes, and virtual personalities** and use intelligent interface, and are seamlessly **intergrated into the information network,** often communicate data associated with users and their environments.

Characteristic 1:

Dynamic & Self Adapting

•IoT devices and systems may have the capability to **dynamically adapt with the changing contexts** and **take actions based on their operating conditions**, user's context, or sensed environment.

Example: Consider a surveillance system comprising of a number of surveillance an adapt their modes (to **normal or infra-red** night modes) based on whether it is day or night.

•Cameras could switch from **lower resolution to higher resolution** modes when any motion is detected and alert nearby cameras to do the same.

Characteristic 2:

Self-Configuring

•IoT devices may have self-configuring capability, allowing a large number of **devices to work together** to provide certain functionality (such as weather monitoring).

•These devices have the **ability configure themselves, setup the networking**, and **fetch latest software upgrades** with minimal manual or user intervention.

Characteristic 3:

Interoperable Communication Protocols

Interoperable: Technology systems and software applications ability to **communicate**, **exchange data**, and **use the information** that has been exchanged

•loT devices may support a number of interoperable communication protocols and can communicate with other devices and also with the infrastructure.

•We describe some of the commonly used communication protocols and models in later sections.

Characteristic 4:

Unique identity

•Each loT device has a **unique identity** and a unique identifier (such as an **IP address** or a **URI**).

(*URI*) is a string of characters used to identify a resource.

•loT systems may have intelligent interfaces which adapt based on the context, allow communicating with users and the environmental contexts.

•IOT device interfaces allow users to query the devices, monitor their status, and control them remotely.

Characteristic 5:

Integrated into Information Network

IoT devices are usually integrated into the information network that allows them to **communicate and exchange data with other devices** and systems.

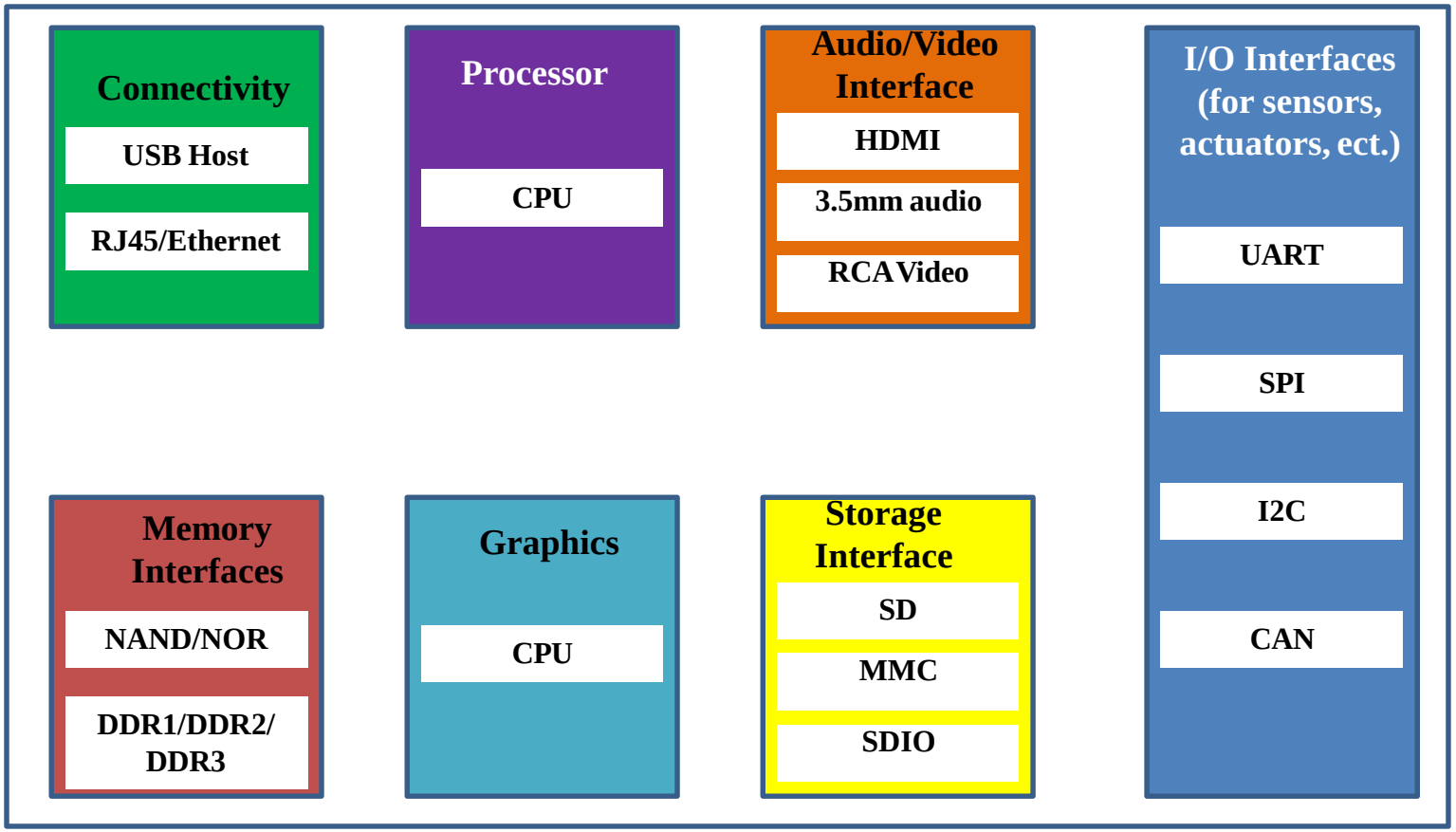
•IoT devices can be **dynamically discovered** in the network, by other devices and/or the network, and have the **capability to describe themselves** (and their characteristics) to other devices or user applications.

Example: A weather monitoring node can describe its monitoring capabilities to another connected node so that they can communicate and exchange data.

•Integration into the information network helps in making loT systems "**Smart** due to the **collective intelligence** of the individual devices in collaboration with the infrastructure.

•Thus, the data from a large number of connected weather monitoring loT nodes can be **aggregated and analyzed to predict the weather**.

1.2 Physical Design of IoT



Block diagram of IoT Device

Communication Protocols

- Communication between electronic devices is like communication between humans.
- Both sides need to speak the same language.
- In electronics, these languages are called **communication protocols**.
- Communication protocols are standards that contains **data exchange rules** and **format** between embedded systems
- UART • I2C • SPI • CAN

UART(Universal Asynchronous Receiver/Transmitter)

- UART is not a communication protocol like SPI and I2C,but a **physical circuit** in microcontroller, or a stand-alone IC.
- It is a computer hardware device for asynchronous serial communication in which **data format** and transmission **speeds** are configurable.
- A UART's main purpose is to **transmit** and **receive serial data**.
- One of the best things about UART is that it only uses **two wires to transmit** data between devices(There is one wire for transmitting data, and one wire to receive data.)
- The transmitting UART converts **parallel data** from a controlling device like a CPU into **serial form**, transmits it in serial to the receiving UART, which then converts the **serial** data back into **parallel data** for the receiving device.

Data Transmission

- A common parameter is the baud rate known as "bps" which stands for bits per second.
- If a transmitter is configured with 9600bps, then the receiver must be listening on the other end at the same speed.
- UART is a serial communication, so bits must travel on a single wire.
- If you wish to send a char (8-bits) over UART, the char is enclosed within **a start and a stop bit**, so to send 8-bits of char data, it would require 2-bit overhead;
- This 10-bit of information is called a **UART frame.**

Why UART?

- **A UART may be used when**

  -High speed is not required

  -An inexpensive communication link between two devices is required.
- **UART communication is very cheap:**
- **Single wire for each direction(and ground wire).**
- **Simple hardware.**
- **In UART communication, two UARTs communicate directly with each other.**

SPI

- Serial Peripheral Interface (SPI) is an interface bus commonly used to send data between microcontrollers and small peripherals such as shift registers, sensors, and SD cards.
- Developed by Motorola to provide full-duplex synchronous serial communication
- It uses separate clock and data lines, along with a select line to choose the device you wish to talk to.
- It is a master – slave type protocol that provides a simple and low cost interface between a microcontroller and its peripherals.

Inter-Integrated Circuit

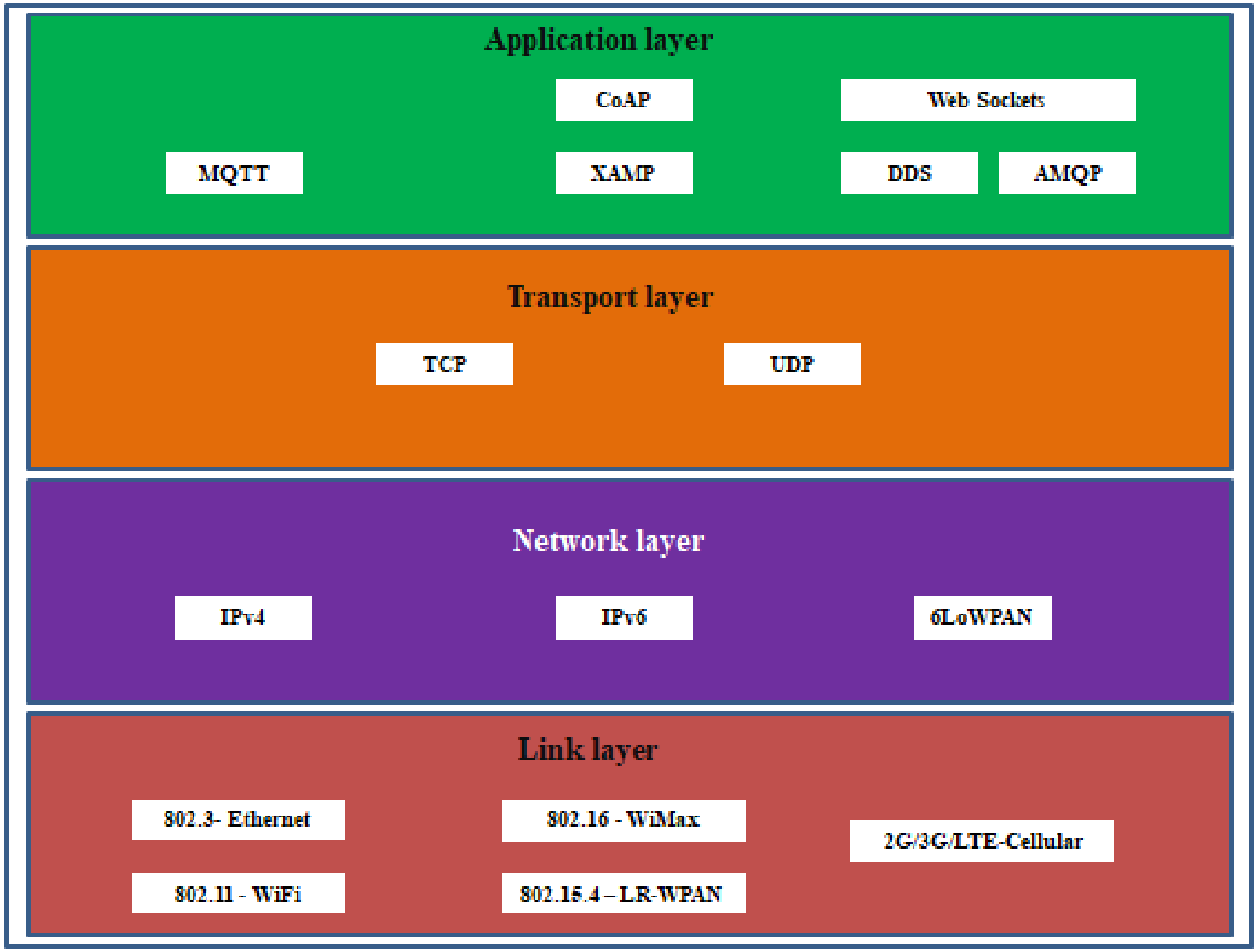
I^2C or I2C (IIC)

- I2C combines the best features of SPI and UARTs. With I2C, you can connect **multiple slaves to a single master** (like SPI) and you can have multiple masters controlling single, or multiple slaves.
- This is really useful when you want to have more than one microcontroller logging data to a single memory card or displaying text to a single LCD.

CAN

- Controller Area Network bus
- It allows microcontroller and devices to communicate without a host computer.
- It is a two **wired half duplex high speed serial** network technology.

1.2.2 IoT Protocols



Link Layer

- Determines **how data is physically sent over** the network's physical layer.

-ex: copper wire, coaxial cable, radio waves

- Scope of LL: Local N/W connections to which host is connected.
- Data exchange-using link layer protocols

Standard	For	Data Rate
IEEE 802.3	Ethernet-Coaxial cable	10MBps to 40GBps
IEEE 802.3i	Ethernet-Twisted pair	
IEEE 802.3j	Ethernet-Fiber	
IEEE 802.11	Wireless LAN	1MBps to 6.7GBps
IEEE 802.16	WifiMax	1.5MBps to 1GBps
IEEE 802.15.4	LR WPAN(Wireless personal area n/w)	40kbps to 250kbps

2G- GSM and CDMA
3G- UMTS and CDMA2000
4G – LTE (Long term evolution)

Data Range: 9.6kbps to 100Mbps

GSM=Global System for Mobile Communications
time division multiple access

CDMA=Code-division multiple access
several transmitters can send information simultaneously over a single communication channel

UMTS=Universal Mobile Telecommunications System

Network Layer

- Responsible for sending datagrams from "Source N/W to Destination N/W"
- Performs **host addressing** and **packet routing**
- Datagram contains source and destination address which is used for routing.
- Addressing schemes: **IPv4 or IPv6**

IPv4

- **Most commonly used**
- **Uses 32-bit address scheme**
- **2^{32} →294967296 address**
- **These address got exhausted in 2011(due to increase in no. of devices)**
- **So moving to IPv6**

IPv6

- Uses 128-bit address scheme
- 2^{128} addresses possible
  340282366920938463463374607431768211456

6LoWPAN

- Low power Wireless Personal Area Network which uses IPv6.
- Brings IP protocol to low power devices

Transport layer

- Provides **end-to-end** message transfer capability independent of underlying N/W
- Transfer capability is set up using TCP or UDP

TCP

•Transmission Control Protocol

• Connection Oriented

•Most widely used by

-web browser

-Email programs

-File Transfer(FTP)

•Provides

-Error detection

-Flow control capability (ensures transmission rate)

-Congestion control capability

Requires initial setup -connection is established and maintained until app program at each end finishes exchange of message

UDP

•User Datagram Protocol

•Connectionless→stateless

•Useful for time sensitive applications where small amount of data will be exchanges

•Doesn't provide guaranteed delivery

Application Layer

- App layer protocols defines how app interface with lower layer protocols to send the data over network.
- App data(in files) **encoded and encapsulated** in transport layer which provides connection or transaction oriented communication over N/W.
- **Port numbers are used** for app addressing

-port 80 for HTTP

-port 22 for SSH ect

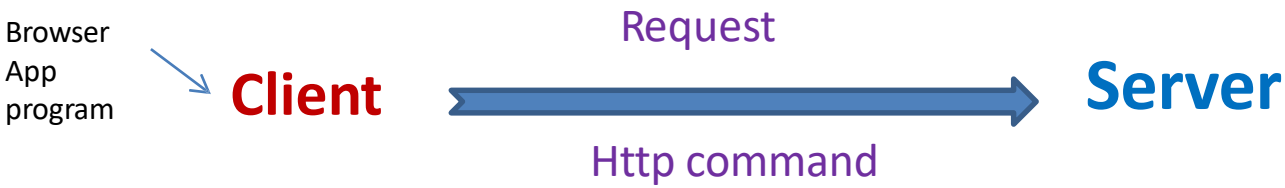
- App layer protocols **enable process-to-process** (service) connections using ports

HTTP: Hypertext Transfer Protocol

- -**Foundation for WWW**

GET → List all resources in the collection
PUT → Replace entire collection with another collection
POST → Create a new resource
DELETE → Delete entire resource

- Includes commands GET, PUT,POST,DELETE ect
- Protocol follows Request-Response Model



- HTTP protocol uses **URI** to identify the HTTP resources

Uniform Resource Identifier (*URI*) is a string of characters used to identify a resource.

CoAP(Constrained Application Protocol)

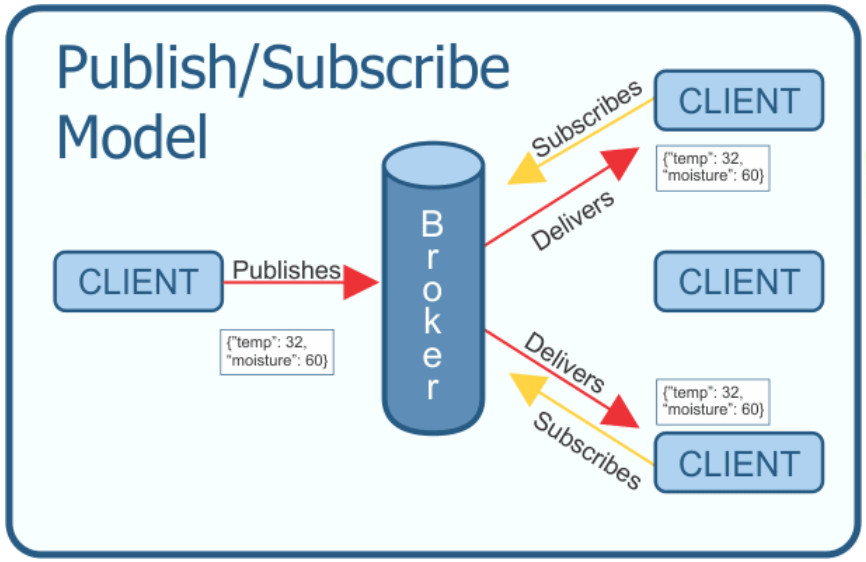
- This is for M2M applications meant for **constrained environment** with **constrained machines** and **networks** (e.g., low-power, lossy networks),
- Uses **Client-Server architecture**
- Clients can communicate with its server using **connectionless datagrams**
- Supports GET,PUT, POST,Delete commands

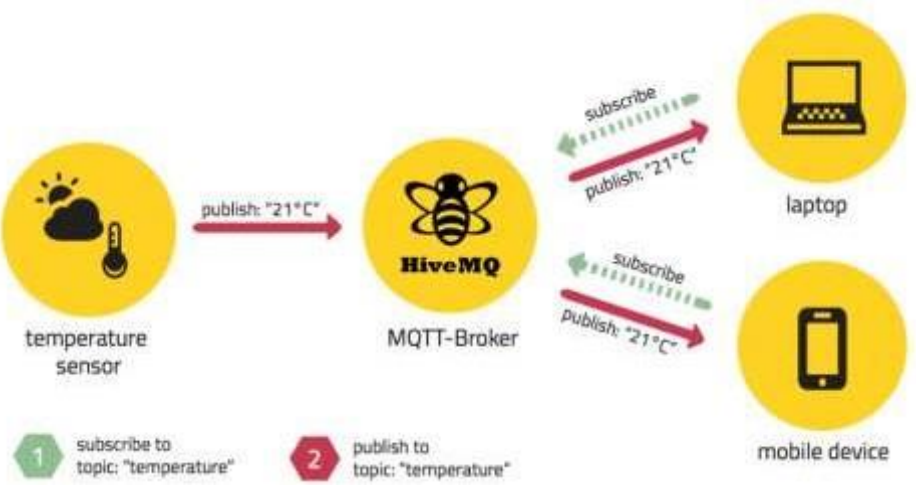
Websockets

- This protocol allows **full duplex communication** over single socket connection
- **Based on TCP**

MQTT(Message Queue Telemetry Transport)

- Light weight messaging protocol based on "**Publish-Subscribe Model**"





XAMP(Extensible Message and Presence Protocol)

- Open XML technology for **real-time communication**, which powers a wide range of applications including **instant messaging, presence, multi-party chat, voice and video calls, and collaboration.**
- The presence indicator tells the servers that you are **online / offline / busy.**

DDS(Data Distribution Service)

- Standard for device to device(M2M) communication
- Uses Publish Subscribe Model with **brokerless** architecture

-i.e Protocol for **Real-time systems**

AMQP(Advanced Message Queuing Protocol)

- Protocol for business messaging
- Supports both **Point-to-Point** Model and **Publish-Subscribe** Model
- Either Broker will distribute the messages or consumers pull the messages

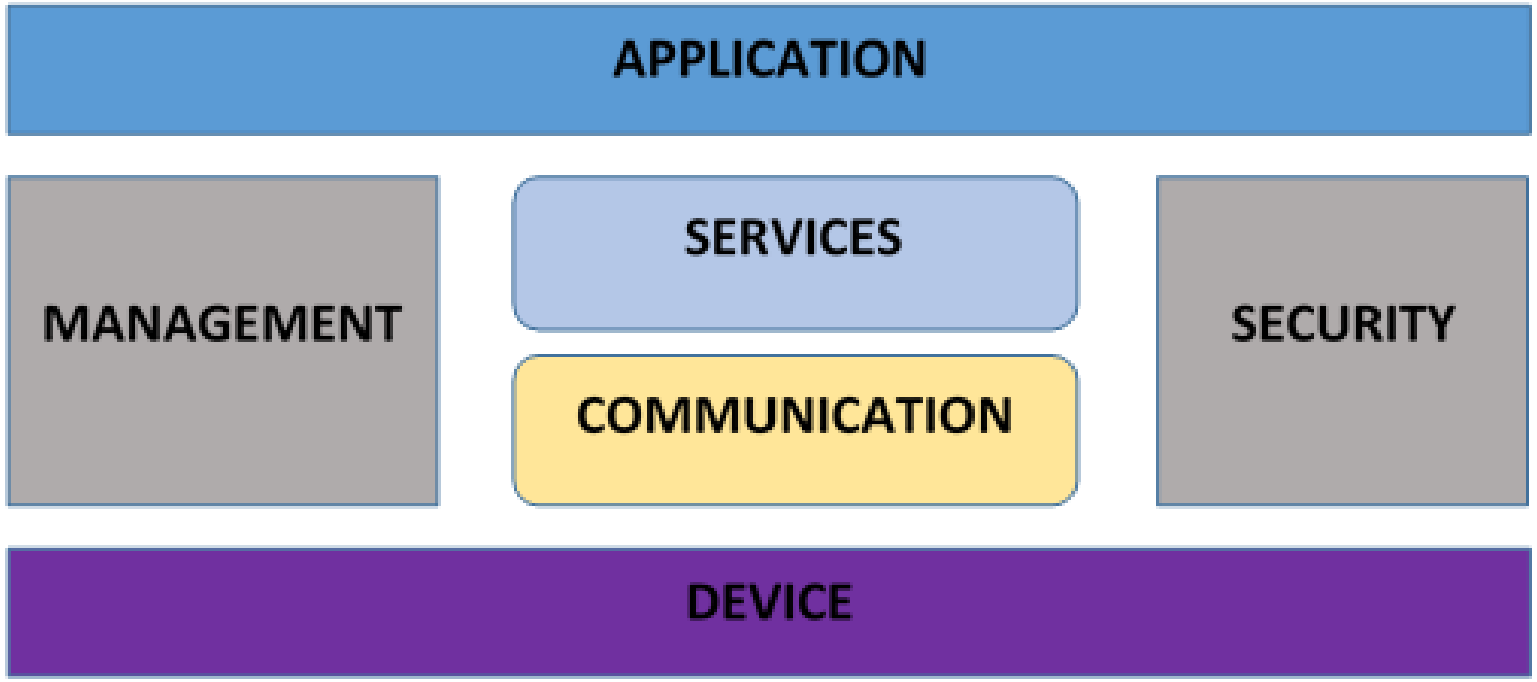
1.3 Logical Design of IoT

- **Abstract representation** of the entities and processes without going into low level specifications of the implementations

1. 1.3.1 **IoT Functional Blocks**
2. 1.3.2 **IoT Communication Models**
3. 1.3.3 **IoT Communication API**

1.3.1 IoT Functional Blocks

An IoT system comprises of a number of functional blocks that provide the system the capabilities for **identification**, **sensing**, **actuation**, **communication**, and **management**.



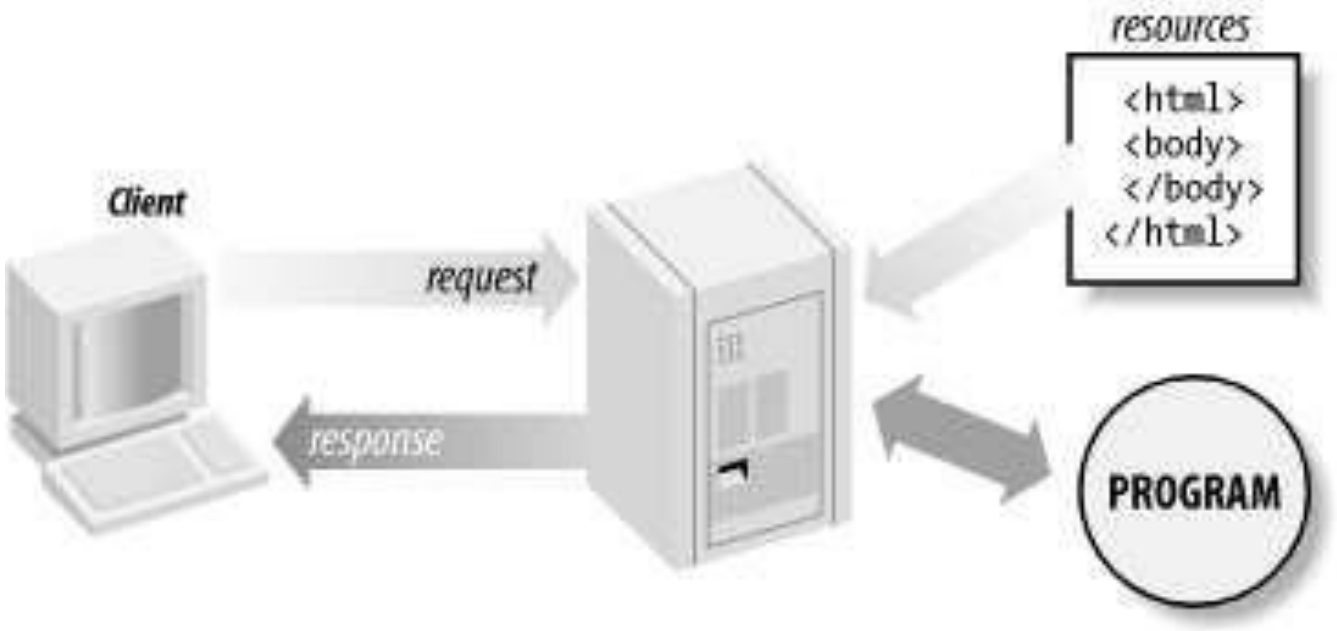
IoT Functional Blocks

1. Device-Provides sensing, actuation, monitoring and control functions
2. Communication- Handles communication for IoT System
3. Services-device monitoring ,device control services, data publishing services
4. Management-Provides various functions to govern the IoT system
5. Security-provides authentication, authorization, message and content integrity and data security
6. Applications-It provides the interface for users to control and monitor the system. It also allows users to view system status and analyze the processed data

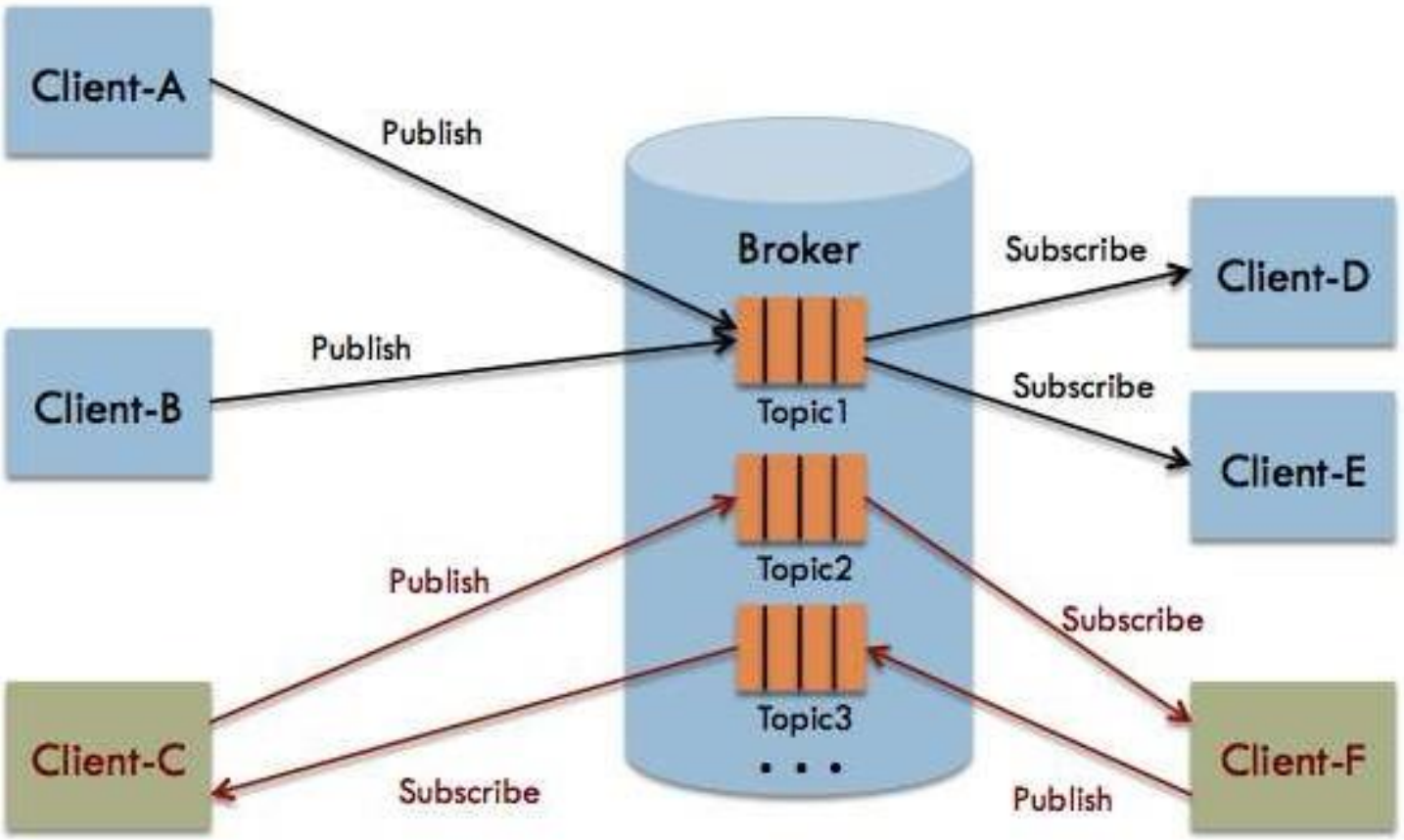
1.3.2 IOT Communication Model

1. Request-Response
2. Publish-Subscribe
3. Push-Pull
4. Exclusive Pair

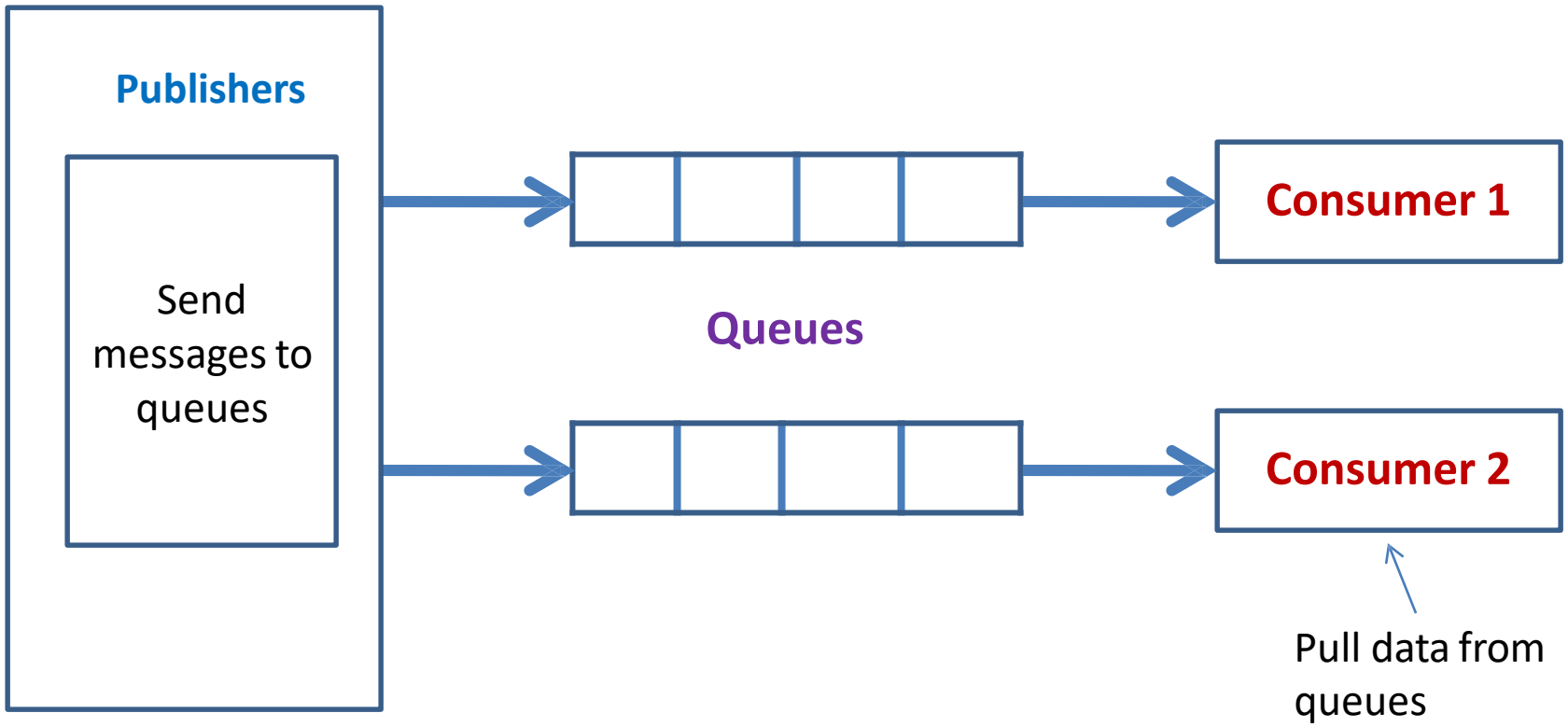
Request-Response



Publish-Subscribe



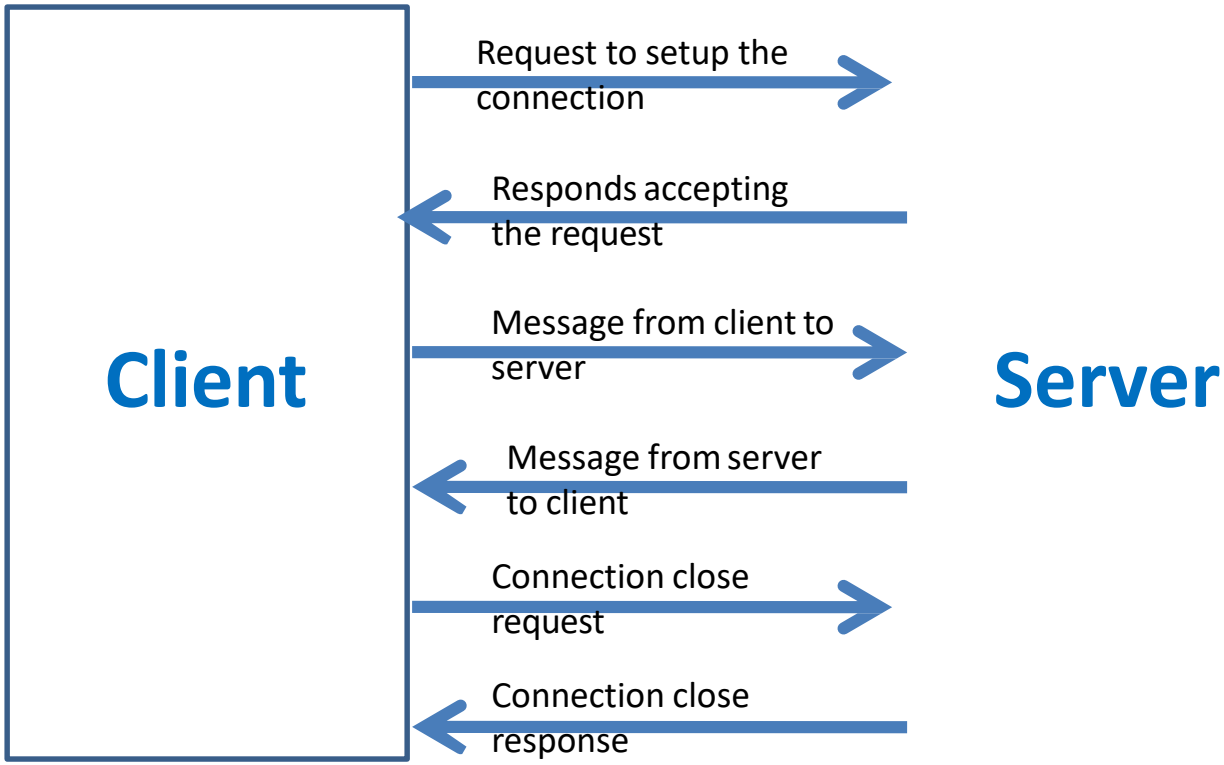
Push-Pull



•Queues use the buffer to avoid mismatch between data rate of Publishers and Subscribers.
•Publishers are not aware of consumers

Exclusive Pair

- Bidirectional and Fully duplex communication



•Statefull communication model: Server is aware of all the open connections.

1.3.3 IOT Communication APIs

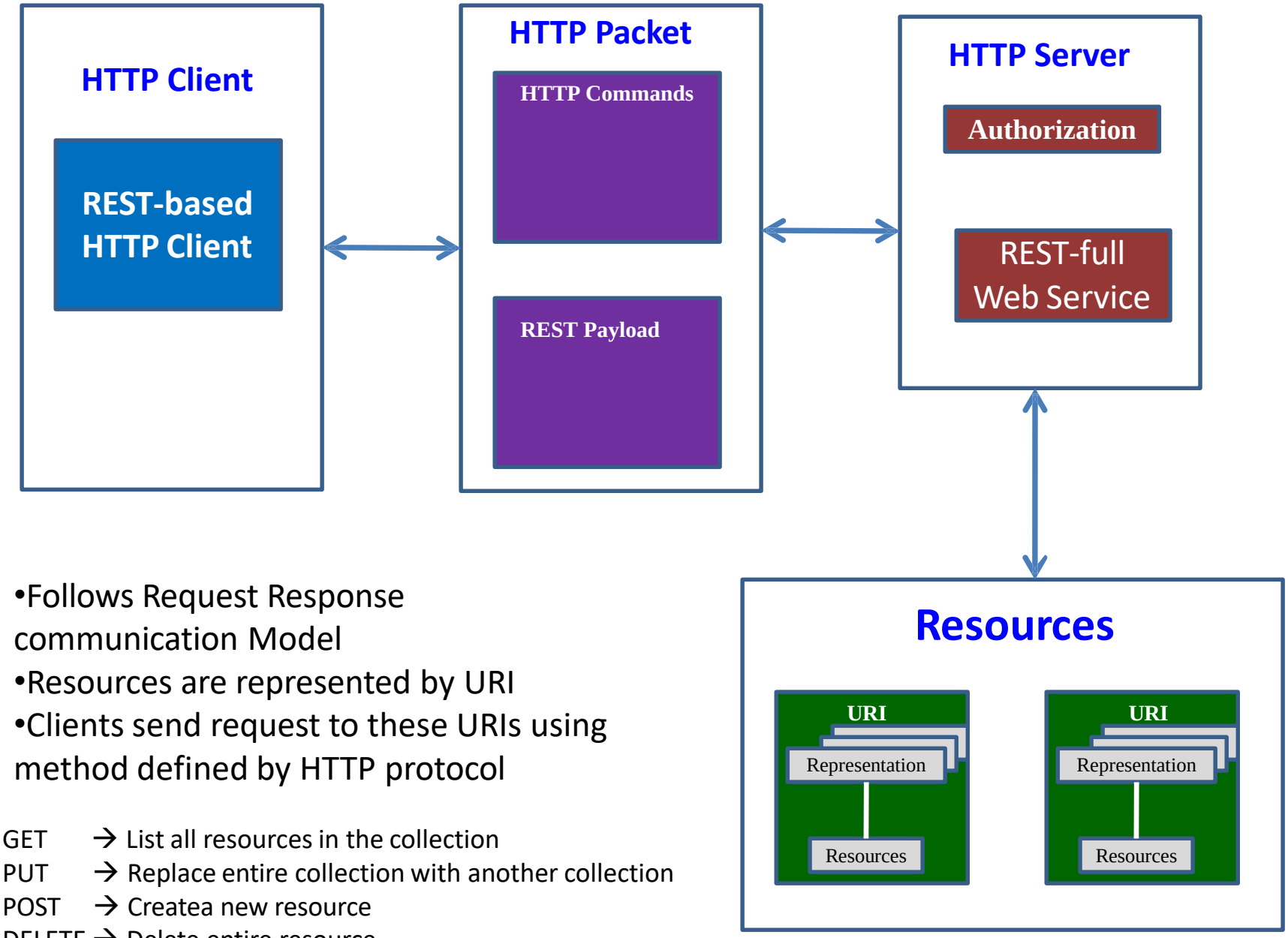
1. REST-based Communication APIs
2. WebSocket-based Communication APIs

API: Set of functions ,protocols, routines and tools used for building application software.

REST-based Communication APIs

- REST stands for **RE**presentational **St**ate Transfer

Communication between Clients and Server using REST API



- Follows Request Response communication Model
- Resources are represented by URI
- Clients send request to these URIs using method defined by HTTP protocol

GET → List all resources in the collection
PUT → Replace entire collection with another collection
POST → Createa new resource
DELETE → Delete entire resource

REST Architecture Constraints

Client-Server

Principal idea is separation of concerns
Ex: Clients should not concerned about storage-->It is concern of Server
Server should not concern about UI --> It is concern of Clients So this separation allows Clients and Server to be independently developed and uploaded

Stateless

Each communication should be independent of others.
Each request from client should include all the info required to understand the request

Cacheable

Cacheable Each response to request is labelled cacheable or non cacheable. If cacheable then client is given right to reuse that response for later for equivalent request.

Layered System

Each component can't see beyond the intermediate layers during interaction.
ex: A Client can't tell if it is directly connected to server or intermediatary along the way

Uniform Interface

Communication between Clients and Server code must be uniform

Code on Demand

Servers can provide executable code for scripts for clients to execute in their context.

WebSocket-based Communication APIs

- Allows bidirectional full duplex communication between Clients and Server
- Follows Exclusive Pair communication model
- Doesn't require a new connection to be set up for each message sent

  -**Connection request: websocket handshake**

  If server supports websocket protocol then server responds to the handshake then server and client can exchange messages in full duplex mode.

No Traffic because no overhead for connection setup