Tutorial 1 : IoT System Design

Dr. Bibhas Ghoshal

Assistant Professor

Department of Information Technology

Indian Institute of Information Technology Allahabad

Outline

- Internet of Things Ecosystem
- Internet of Things Architecture – Four Layered Structure
- Logical Design of IoT
- IoT Levels and Deployment Techniques
- Communication Models
  - Request Response
  - Publish Subscribe
  - Exclusive Pair
  - REST based APIs
- IoT Design Methodology
- Case Study : Designing a Home Automation System

Internet of Things

Internet of things

From Wikipedia, the free encyclopedia

The **Internet of things** (**IoT**) describes the network of physical objects—"things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the Internet.[1][2][3][4]

Things have evolved due to the convergence of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded systems.[1] Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", including devices and appliances (such as lighting fixtures, thermostats, home security systems and cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers. IoT can also be used in healthcare systems.[5]

Things, People and Cloud Services get connected via Internet to enable new use cases and business models

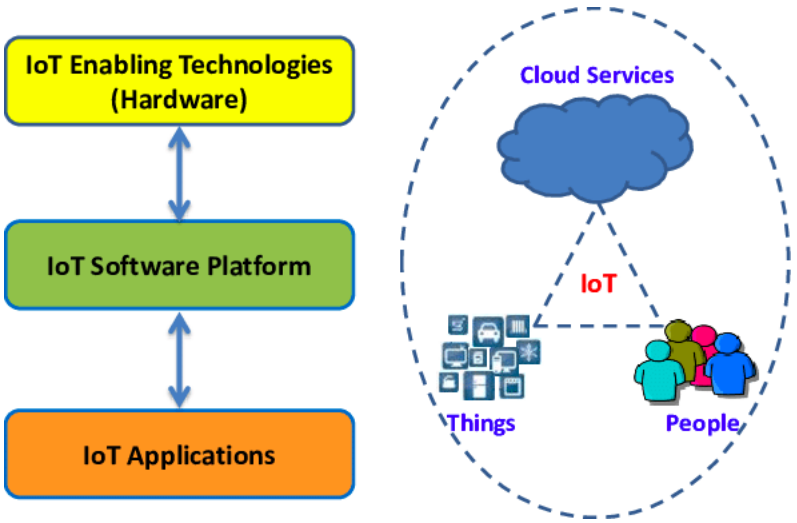
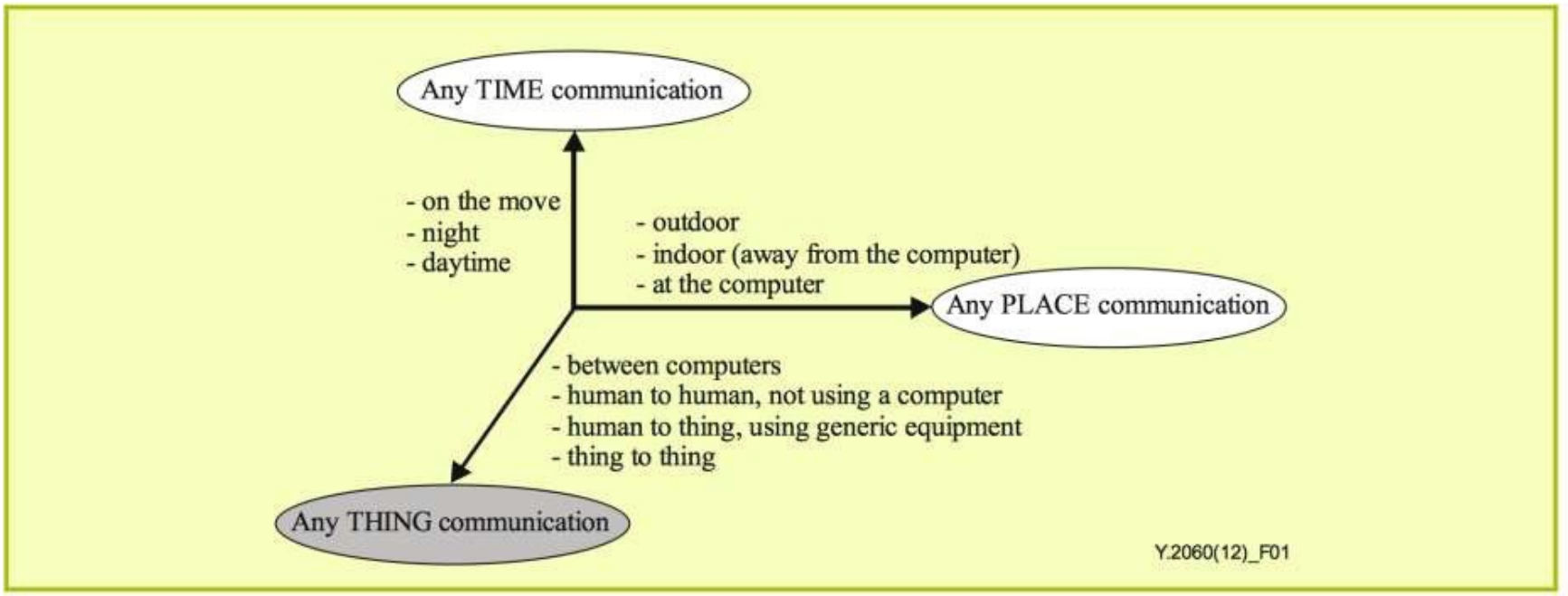


Figure source : Iman Khajenisiri et.al., A Review of Internet of Things Solution for Intelligent Energy Control in Buildings for Smart City Applications, Energy Procedia, Volume 111, 2017

Internet of Things : Any Place / Time / Thing



Source: Recommendation **ITU-T Y.2060**

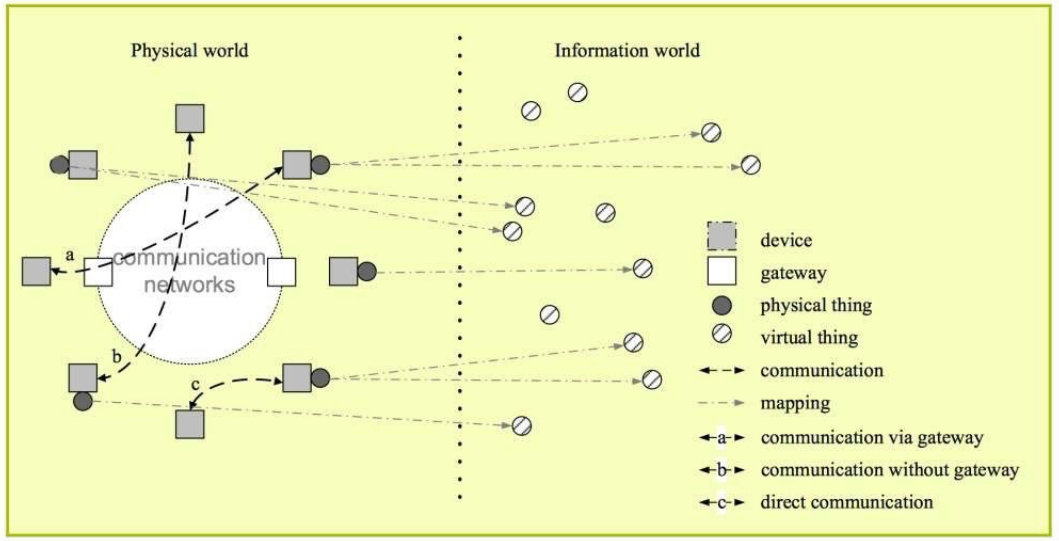
Physical and Virtual Things

Physical things exist in the physical world and are capable of being sensed, actuated and connected.

Ex: electrical and mechanical equipments; goods; robots etc.

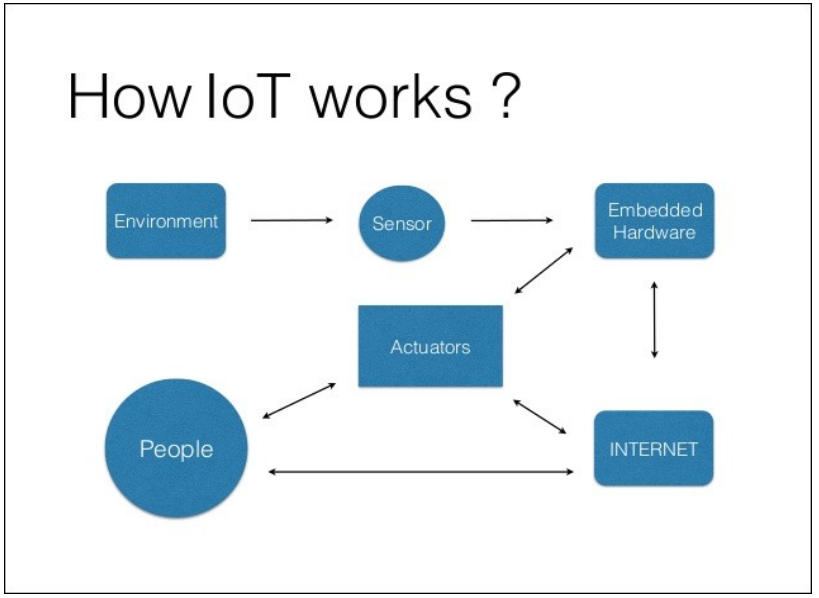
Virtual things exist in the information world and are capable of being stored, processed and accessed.

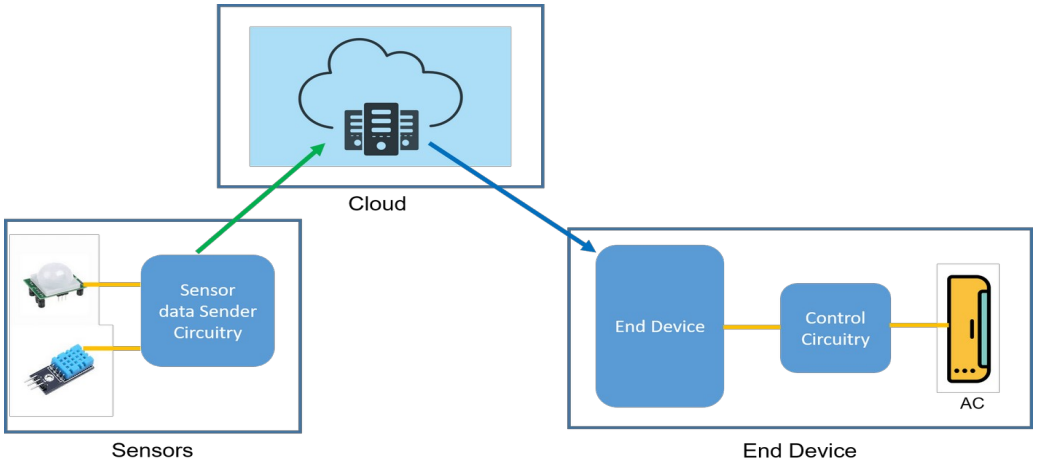
Ex: multimedia content and application software.



Source: Recommendation **ITU-T Y.2060**

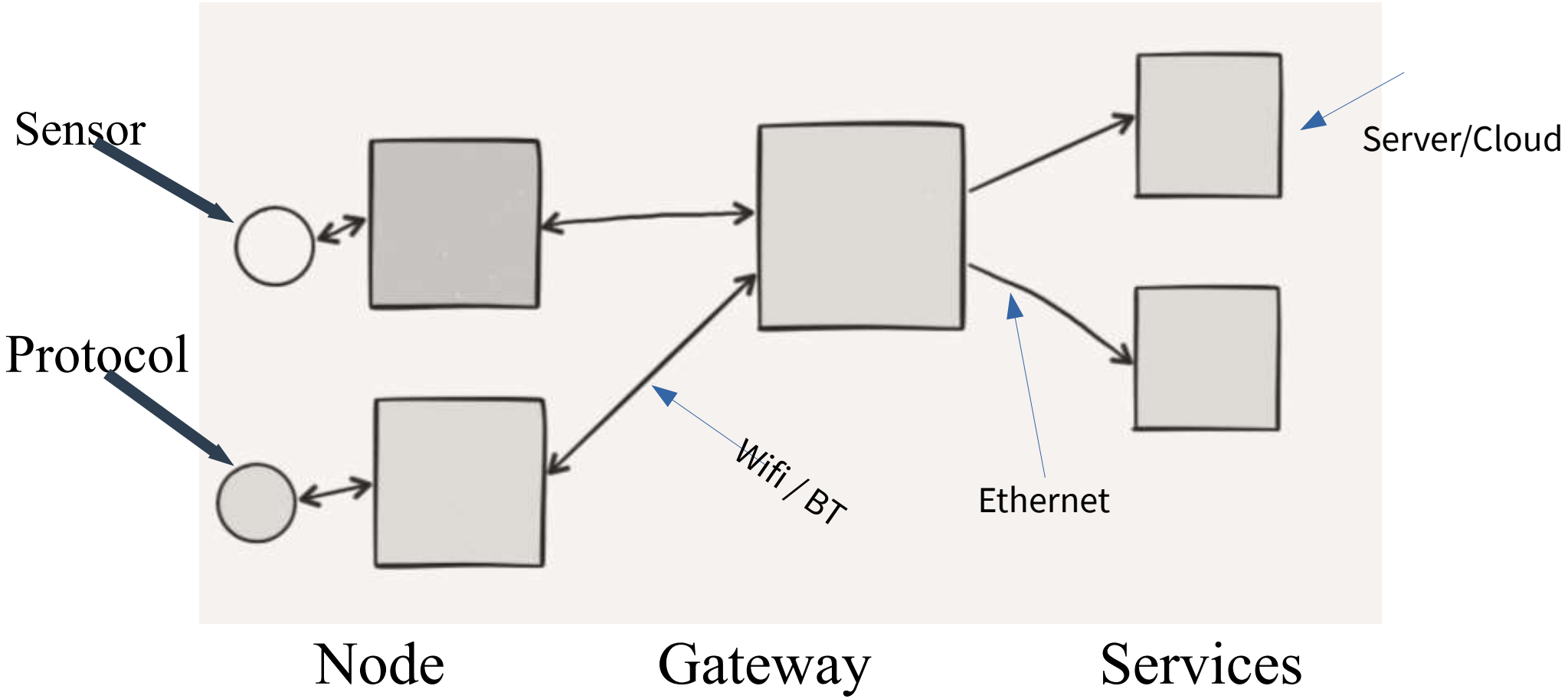
IoT – Basic Idea



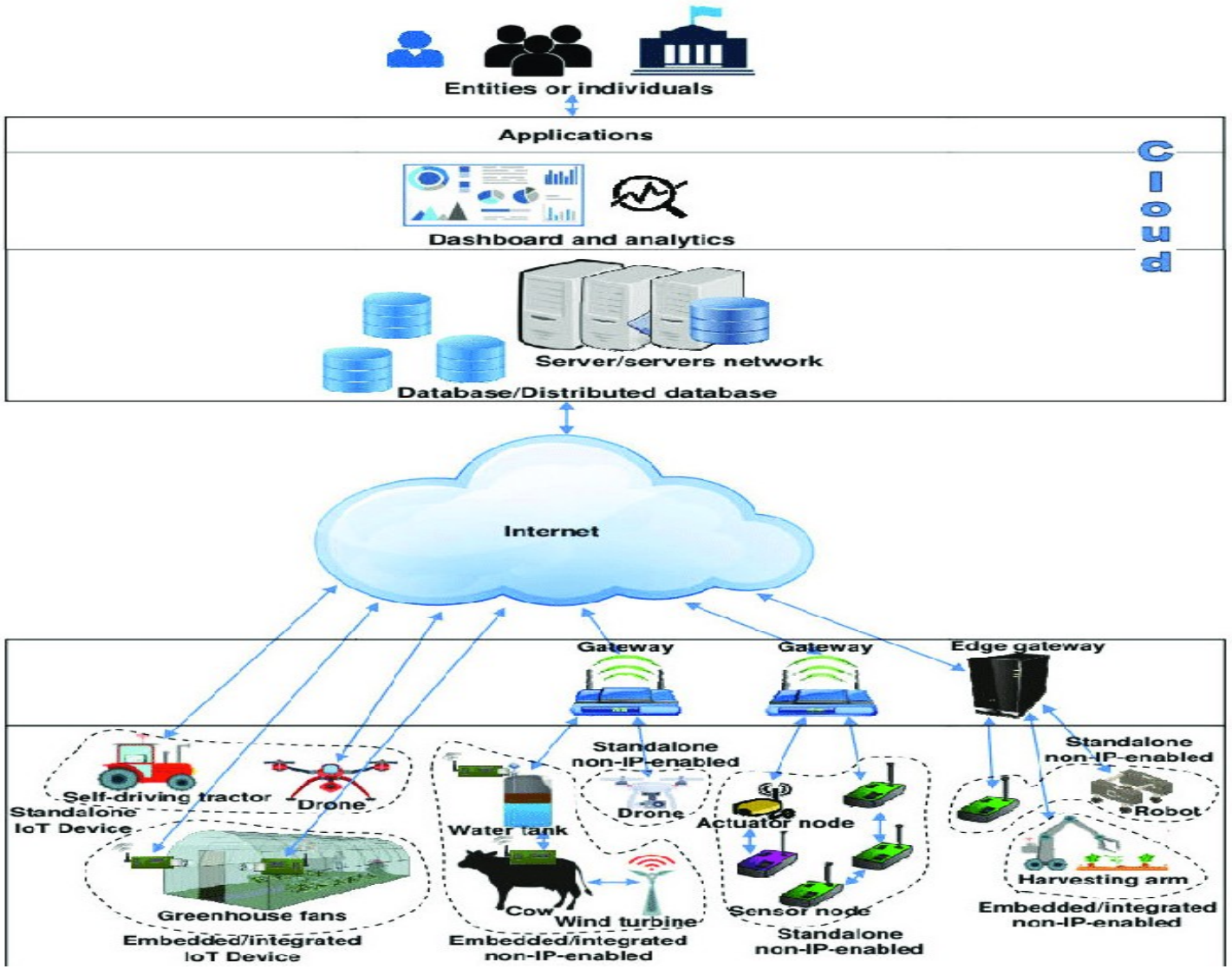


IoT System Design

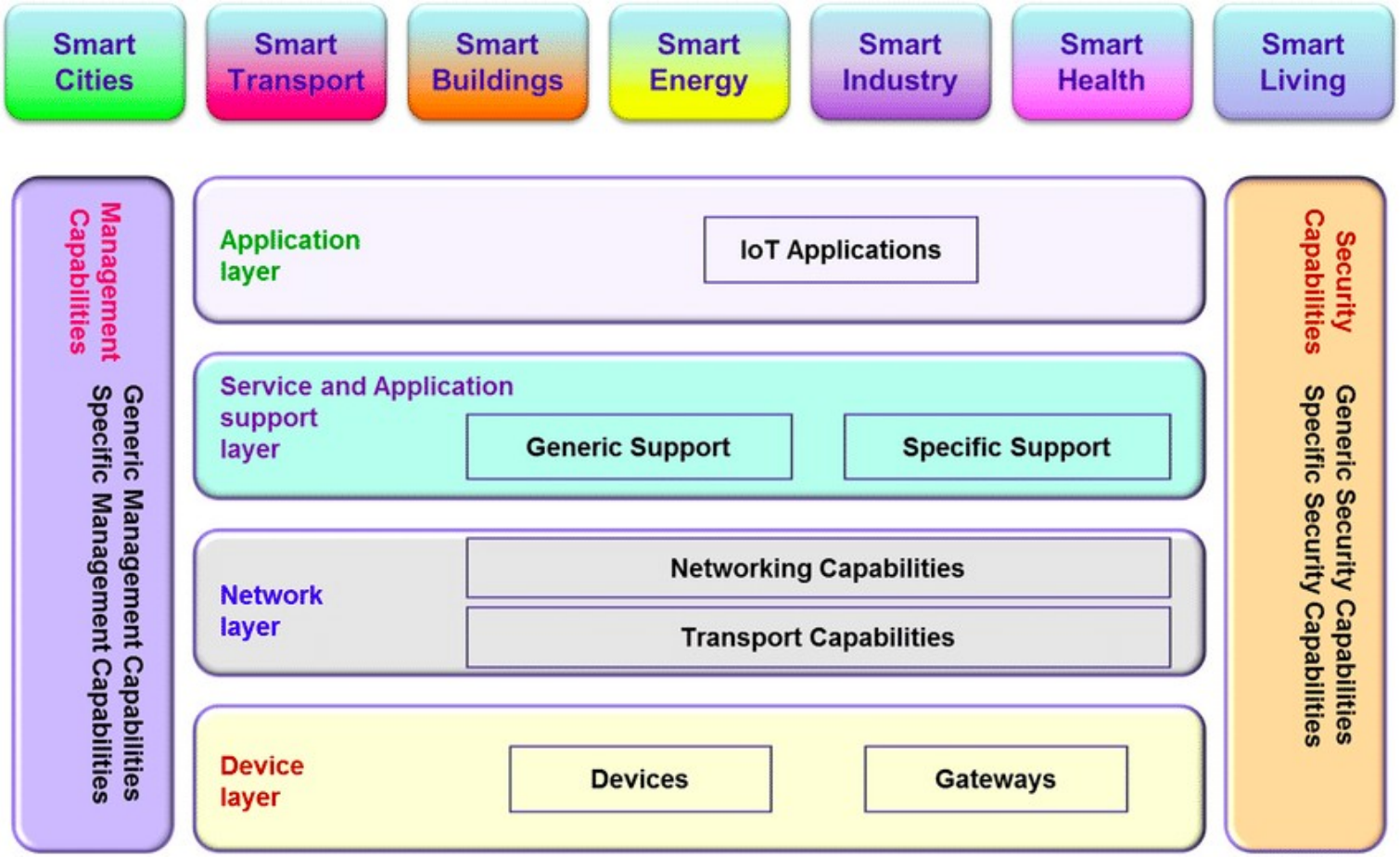
IoT architecture : Simplified model



IoT Ecosystem



The Four Layered IoT Architecture

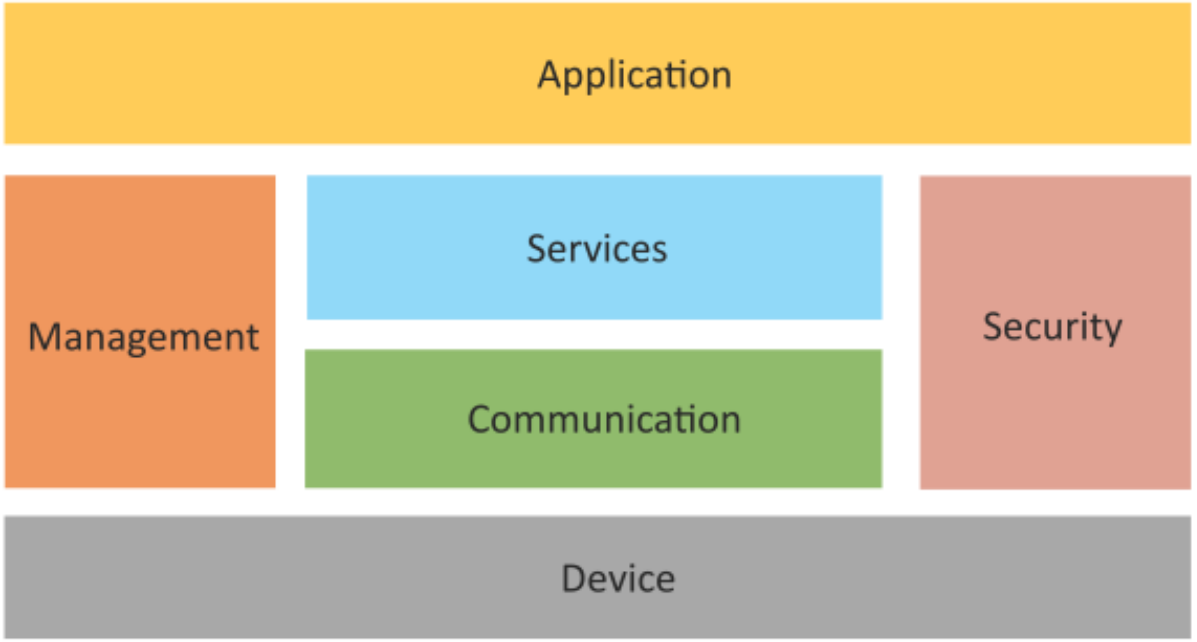


Source : ITU-T

Logical Design of IoT Based Systems

Logical design of an IoT system refers to an abstract representation of the entities and processes without going into the low-level specifics of the implementation.

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



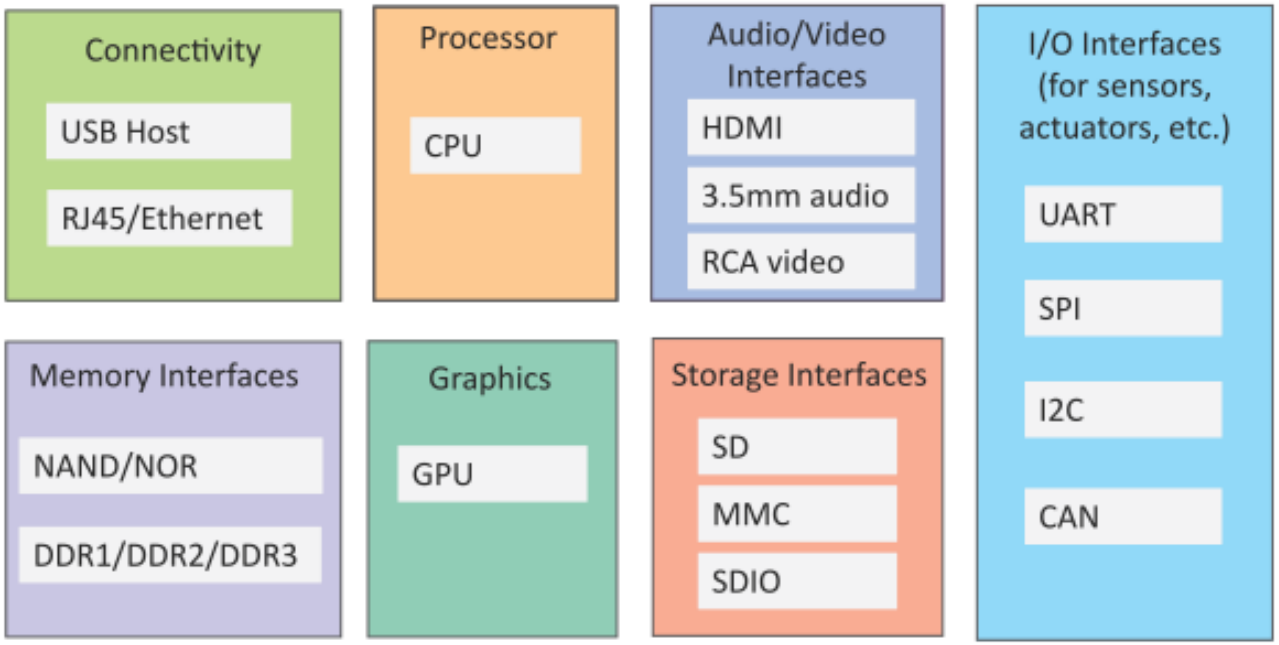
Logical Design of IoT

Source: Book website: http://www.internet-of-things-book.com

IoT System Design : IoT Levels and Deployment

An IoT system comprises the following components :

Devic e : An IoT device allows identification, remote sensing, actuating and remote monitoring capabilities.

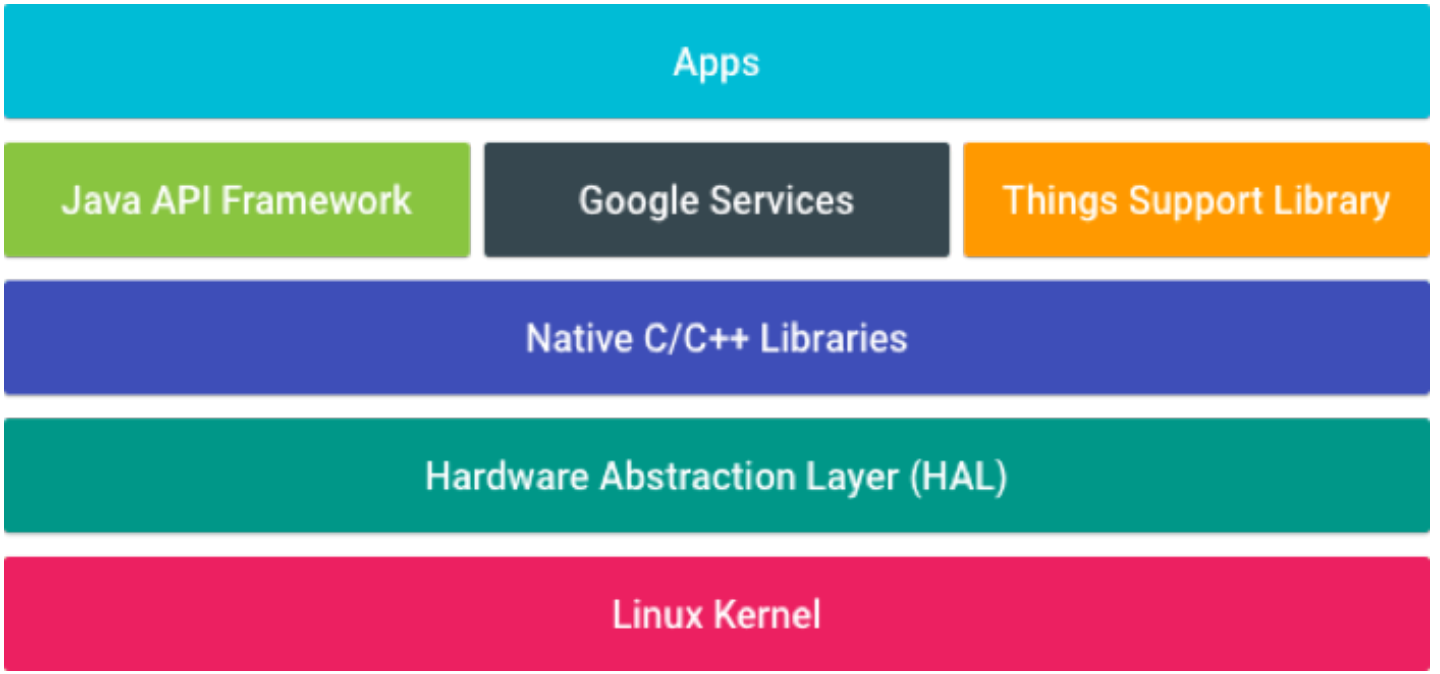


Source: Book website: http://www.internet-of-things-book.com

IoT System Design : IoT Levels and Deployment

Resourc es : Resources are software components on the IoT device for accessing, processing and storing sensor information, or for controlling actuators connected to the device. Resources also include the software components that enable network access for the device.

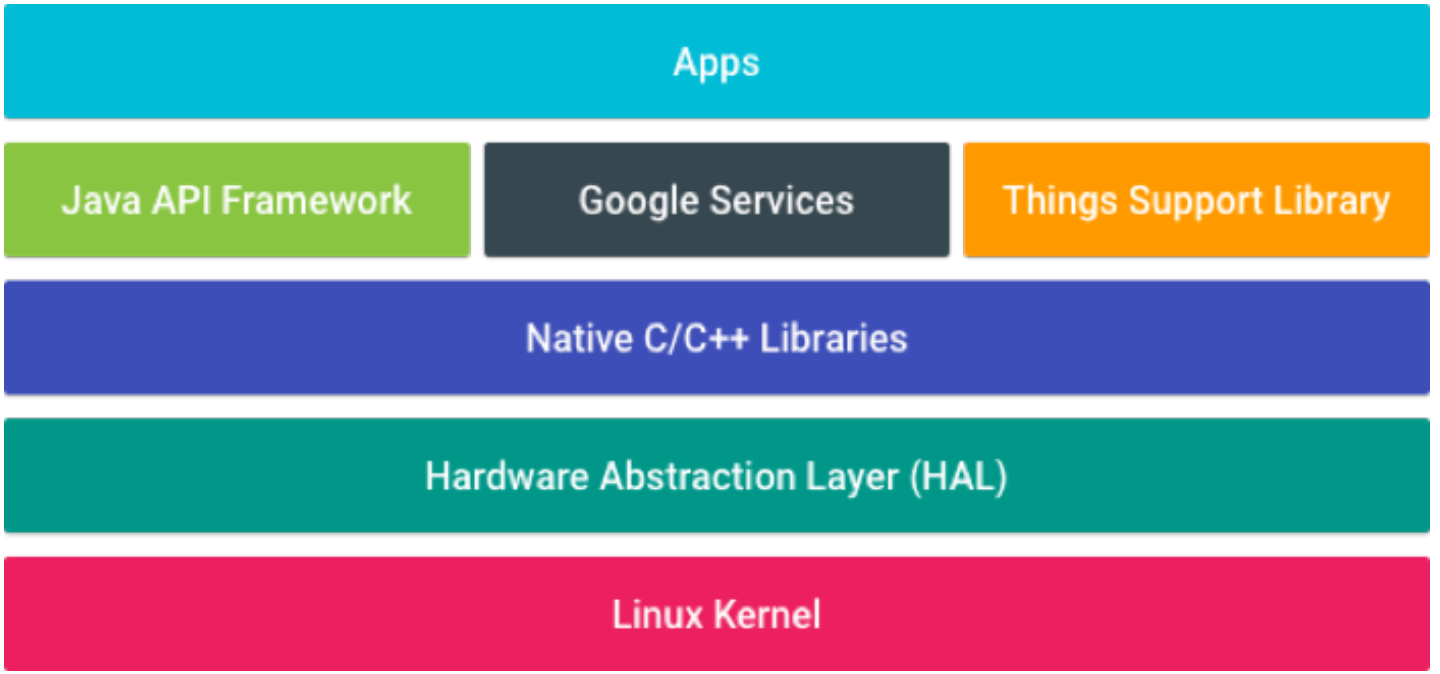
Controller Service : Controller service is a native service that runs on the device and interacts with the web services. Controller service sends data from the device to the web service and receives commands from the application (via web services) for controlling the device.



IoT System Design : IoT Levels and Deployment

Database : Database can be either local or in the cloud and stores the data generated by the IoT device.

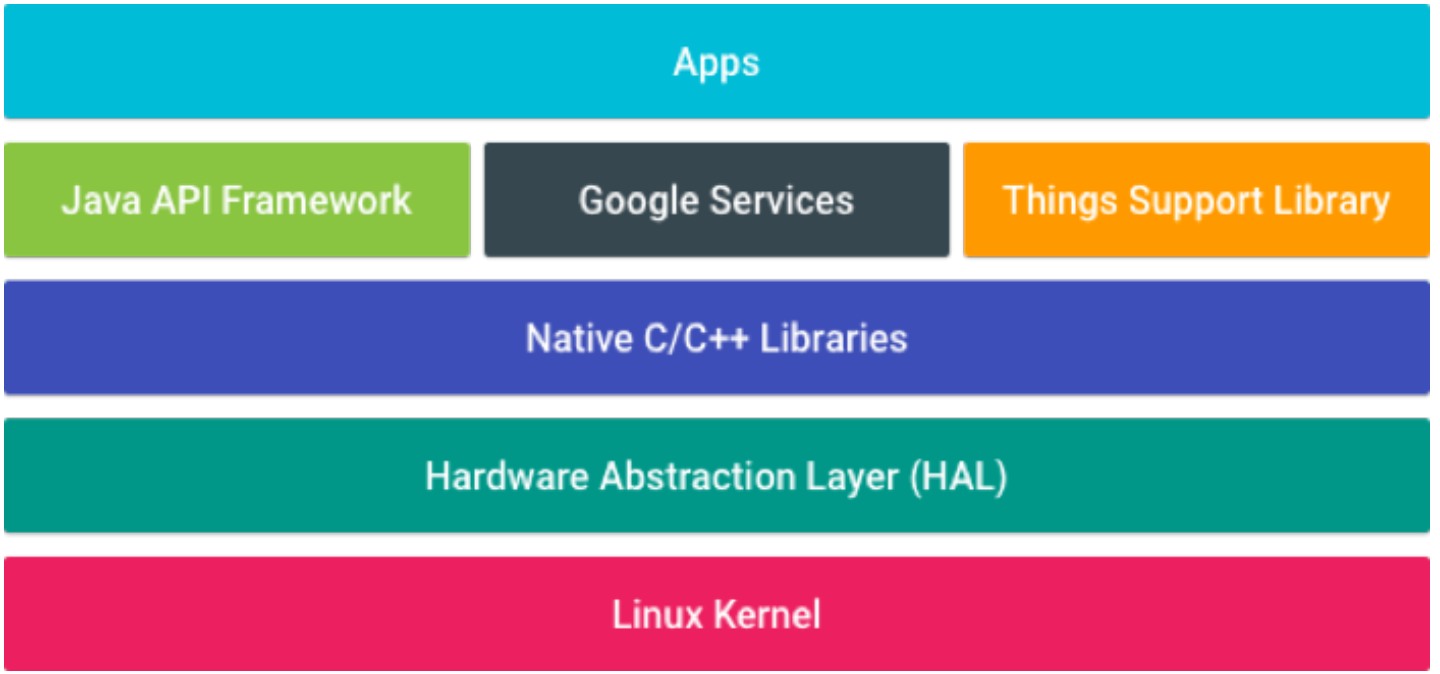
Web Service : Web services serve as a link between the IoT device, application, database and analysis components. Web service can be implemented using HTTP and REST principles (REST service) or using the WebSocket protocol (WebSocket service).



IoT System Design : IoT Levels and Deployment

Analysis Component : This is responsible for analyzing the IoT data and generating results in a form that is easy for the user to understand.

Application : IoT applications provide an interface that the users can use to control and monitor various aspects of the IoT system. Applications also allow users to view the system status and the processed data.



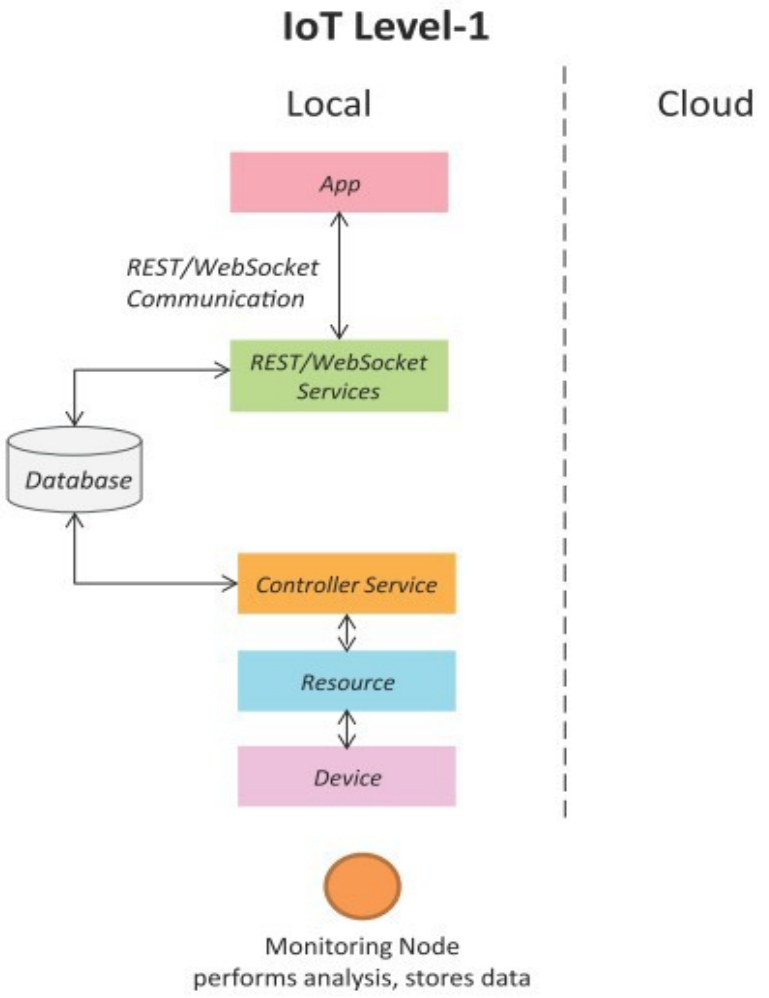
IoT Level - 1

IoT Level - 1 has a **single node** that performs sensing and/or actuation and hosts applications.

Data is stored at the node.

Application runs at the node

They are suitable for modelling low cost solutions where **data is not big and analysis requirements are not computationally intensive**



Source: Book website: http://www.internet-of-things-book.com

IoT Level - 1 : Smart Home

Sensors used : Light, Temperature, Motion, Camera

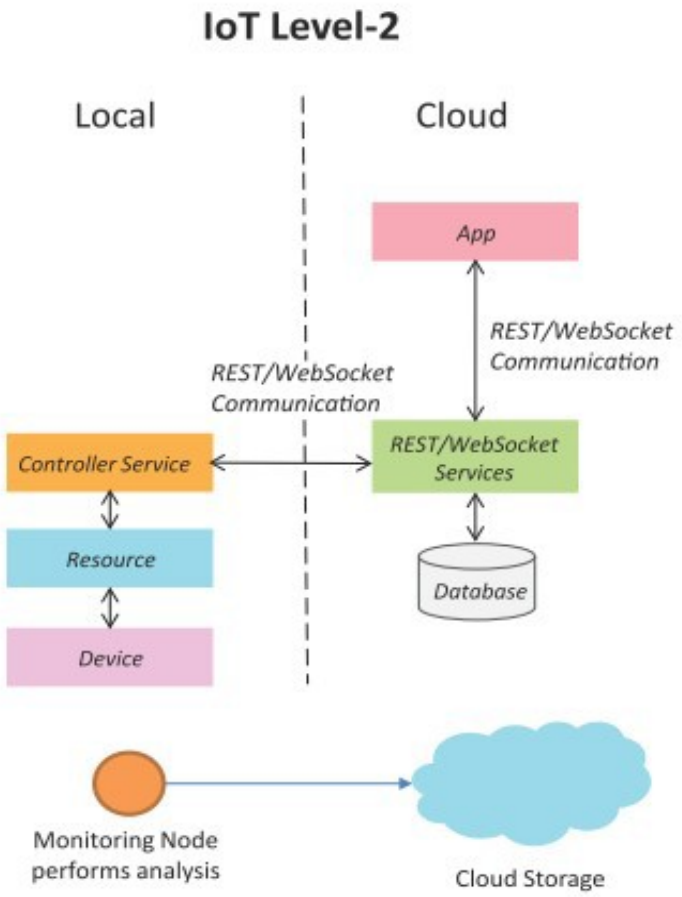
Source: Internet

IoT Level - 2

IoT Level - 2 has a single node that performs sensing and/or actuation and local analysis.

Data is stored in the cloud and application is cloud based

They are suitable for modelling where **data is big** However, **analysis requirements are not computationally intensive** and can be done locally



Source: Book website: http://www.internet-of-things-book.com

IoT Level - 2 : Smart Irrigation

Sensors used : Temperature, Humidity, Pressure

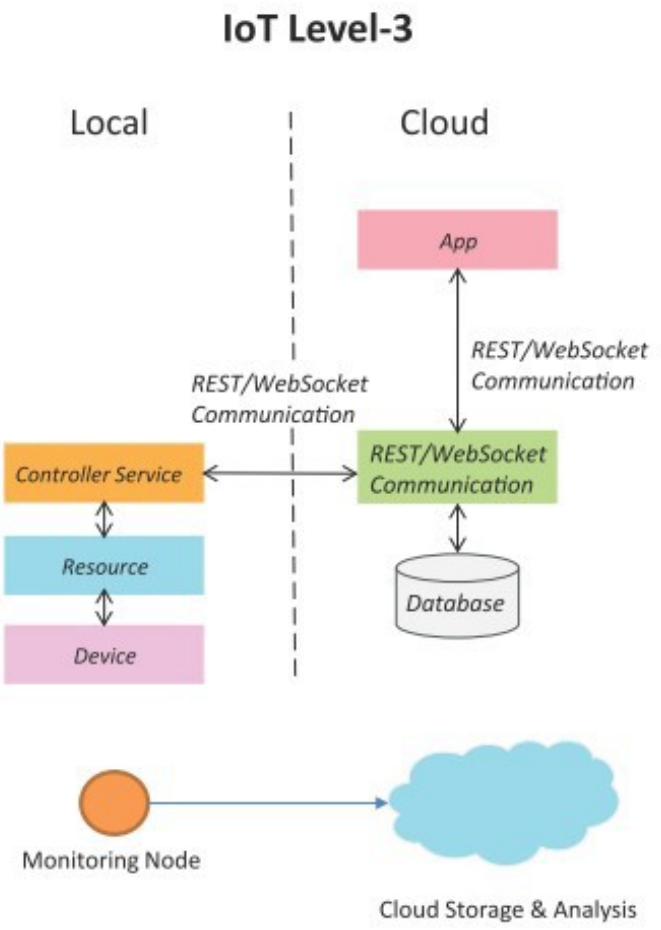
Source: Internet

IoT Level - 3

IoT Level - 3 has a single node that performs sensing and/or actuation and local analysis.

Data is stored in the cloud and application is cloud based

They are suitable for modelling where **data is big** However, **analysis requirements are computationally intensive**



Source: Book website: http://www.internet-of-things-book.com

IoT Level - 3 : Tracking Package Delivery

Sensors used : Gyroscope and Accelerometer

Figure Source: Internet

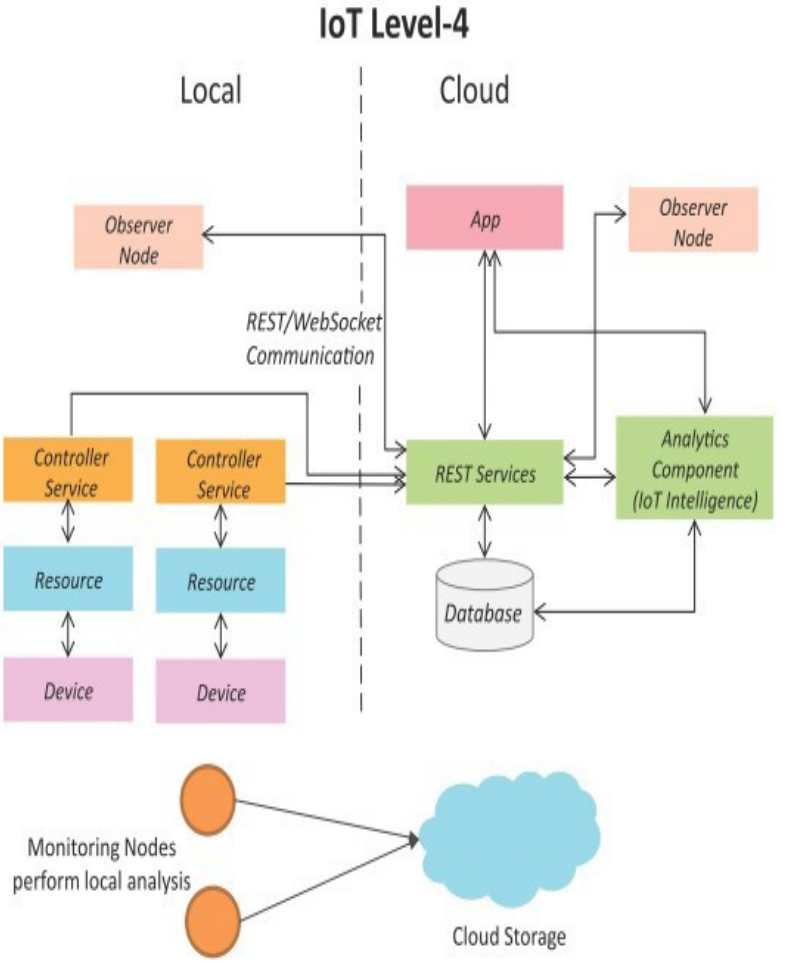
IoT Level - 4

IoT Level - 4 has a **multiple nodes** that performs local analysis

Data is stored in the cloud and application is cloud based.

Local and cloud based **observer nodes** which can subscribe to and receive information collected in the cloud

They are suitable where multiple nodes are Required, **data is big and analysis requirements are computationally intensive**



Source: Book website: http://www.internet-of-things-book.com

IoT Level – 4 : Noise Monitoring System

Sensors used : Sound Sensors

Figure Source: Book website: Norsonic Asia

IoT Level - 5

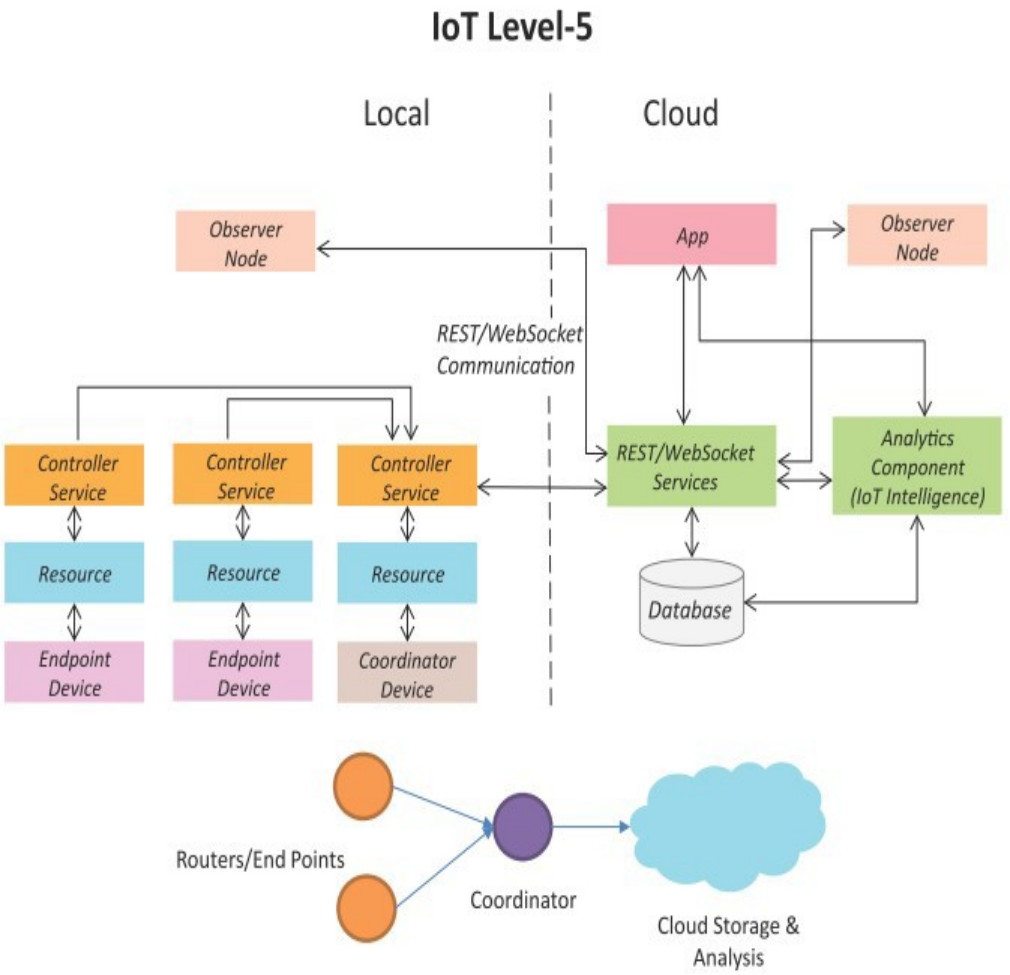
IoT Level - 5 has a **multiple end nodes** and one **co-ordinator node**

End node performs sensing and actuation

Collector node collects data from end nodes and send it to the cloud

Data is stored in the cloud and application is cloud based.

Level 5 IoT systems are suitable for solutions based on wireless sensor network in which **data is big and analysis requirements are computationally intensive**



Source: Book website: http://www.internet-of-things-book.com

IoT Level – 5 : Forest Fire Detection System

- Sensors used : temperature, smoke, weather, slope of earth, wind speed, speed of fire, flame length

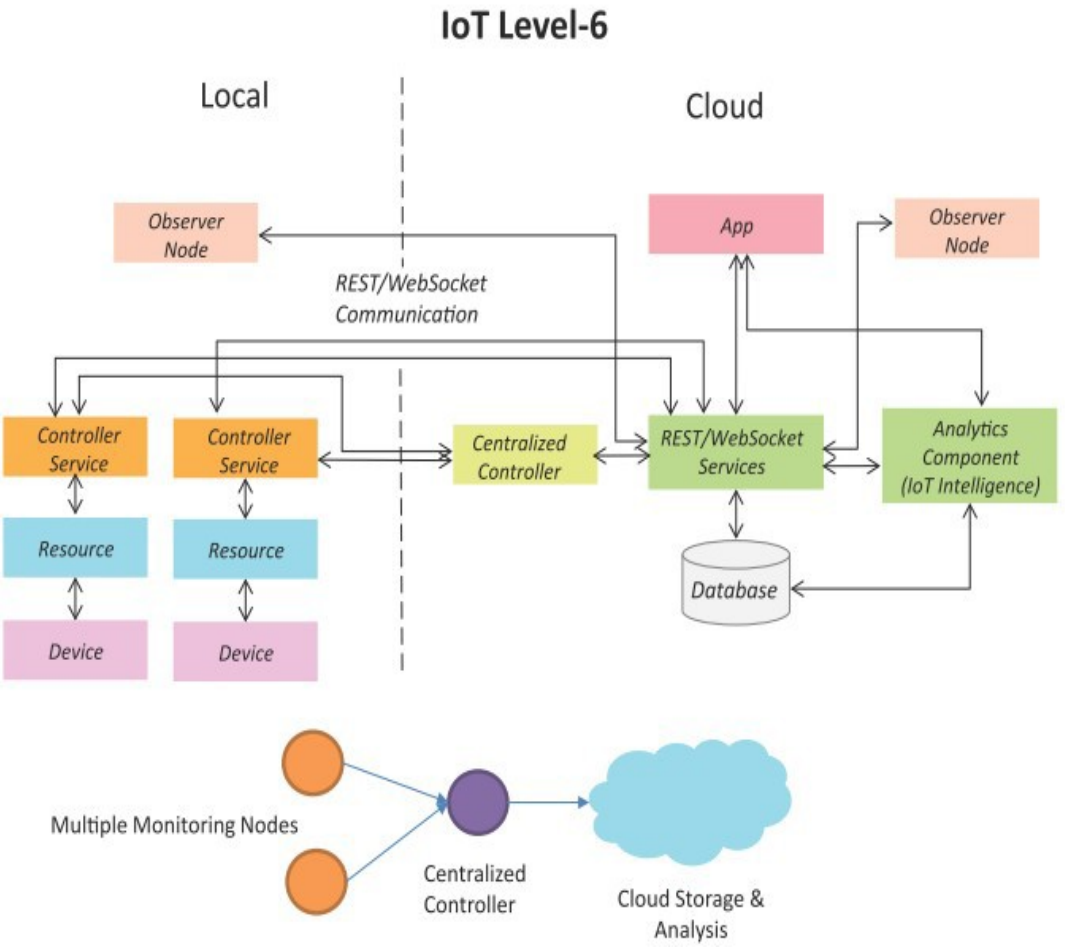
IoT Level - 6

IoT Level - 6 has a multiple independent end Nodes that perform sensing and / or actuation and send data to the cloud.

Data is stored in the cloud and application is cloud based.

The analytics component analyses data and stores it in the cloud database

The results are visualized with the cloud based application. **The centralized controller is aware of the status of all end nodes and sends cor commands to the nodes**



Source: Book website: http://www.internet-of-things-book.com

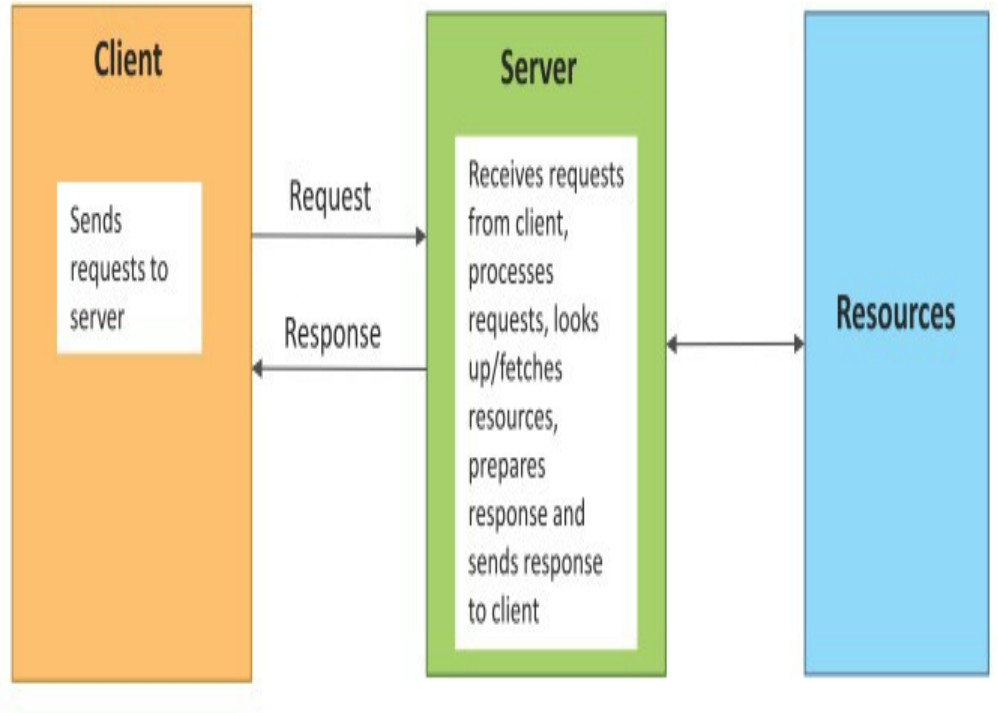
IoT Level – 6 : Weather Monitoring Station

- Sensors used : wind speed direction, solar radiation, temperature, relative humidity, precipitation, snow depth, barometric pressure,soil moisture

Communication Models : Request - Response

Request–Response is a communication model in which the client sends requests to the server and the server responds to the requests

When the server receives a request, it decides how to respond, fetches the data, retrieves resource representations, prepares the response and then sends the response to the client



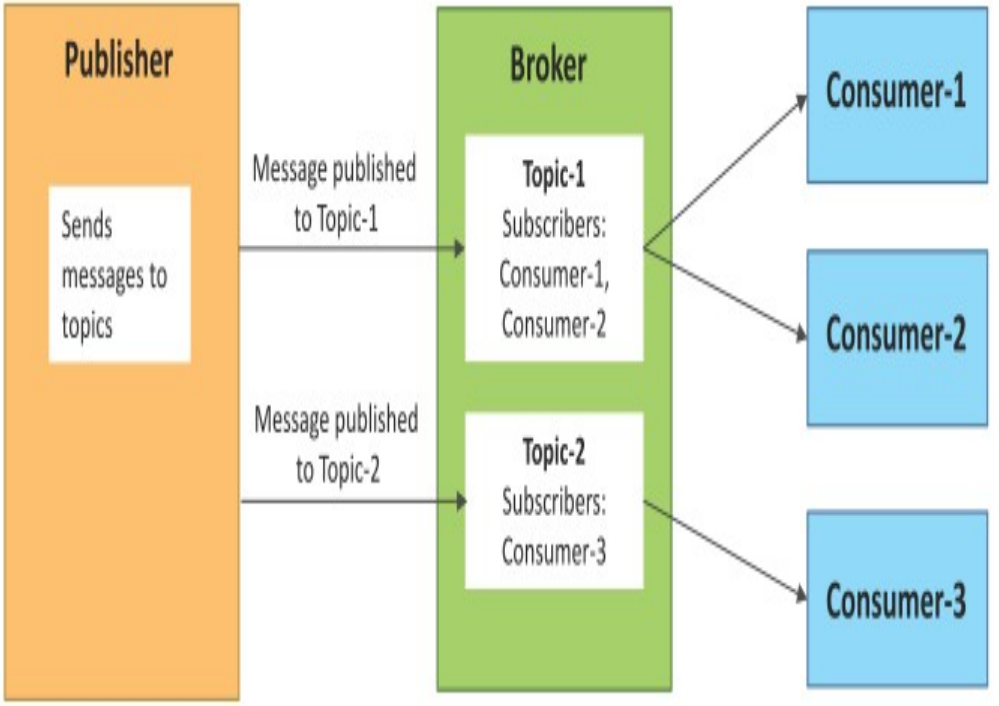
Source: Book website: http://www.internet-of-things-book.com

Communication Models : Publish- Subscribe

Publish–Subscribe is a communication model that involves publishers, brokers and consumers

Publishers are the source of data. Publishers send the data to the topics which are managed by the broker. Publishers are not aware of the consumers.

Consumers subscribe to the topics which are managed by the broker. When the broker receives data for a topic from the publisher, it sends the data to all the subscribed consumers.



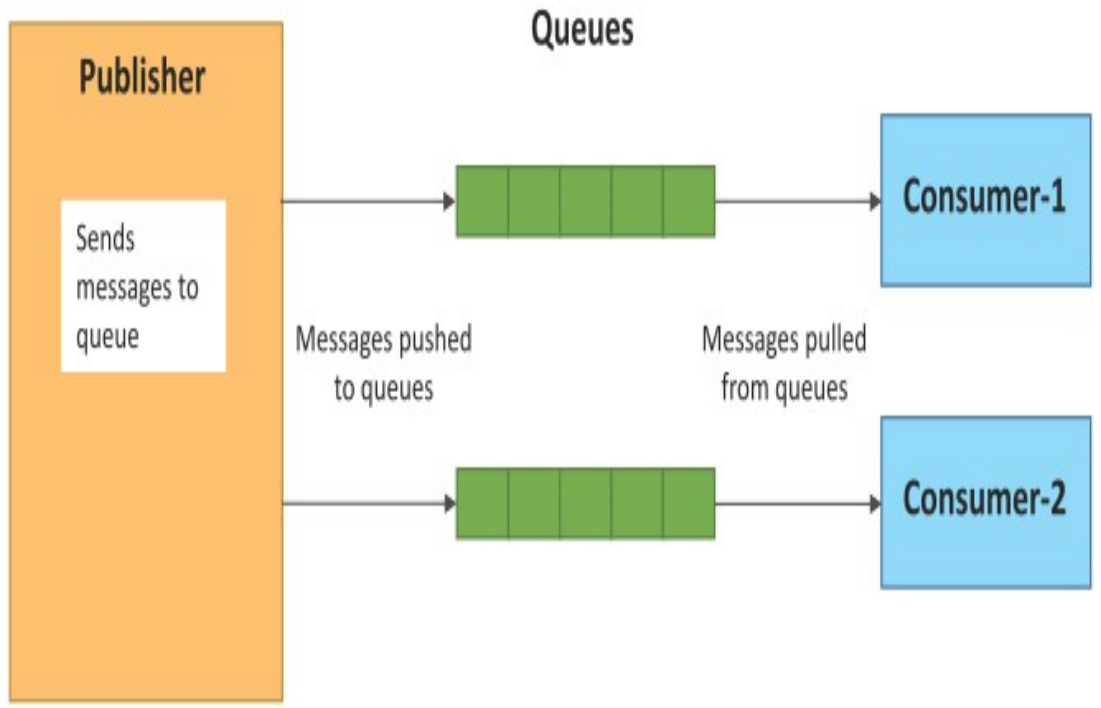
Source: Book website: http://www.internet-of-things-book.com

Communication Models : Push - Pull

Push–Pull is a communication model in which the data producers push the data to queues and the consumers pull the data from the queues. Producers do not need to be aware of the consumers

Queues help in decoupling the messaging between the producers and consumers.

Queues also act as a buffer which helps in situations when there is a mismatch between the rate at which the producers push data and the rate at which the consumers pull data.



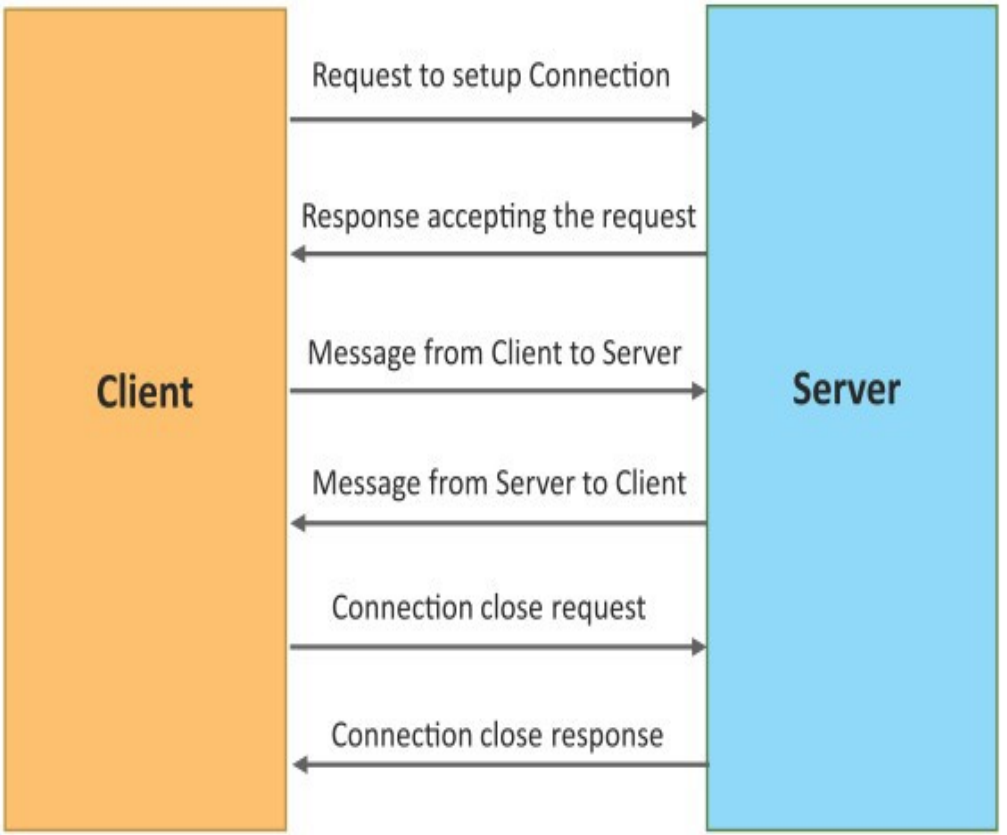
Source: Book website: http://www.internet-of-things-book.com

Communication Models : Exclusive Pair

Exclusive Pair is a bidirectional, fully duplex communication model that uses a persistent connection between the client and the server.

Once the connection is set up it, remains open until the client sends a request to close the connection.

Client and server can send messages to each other after connection setup.



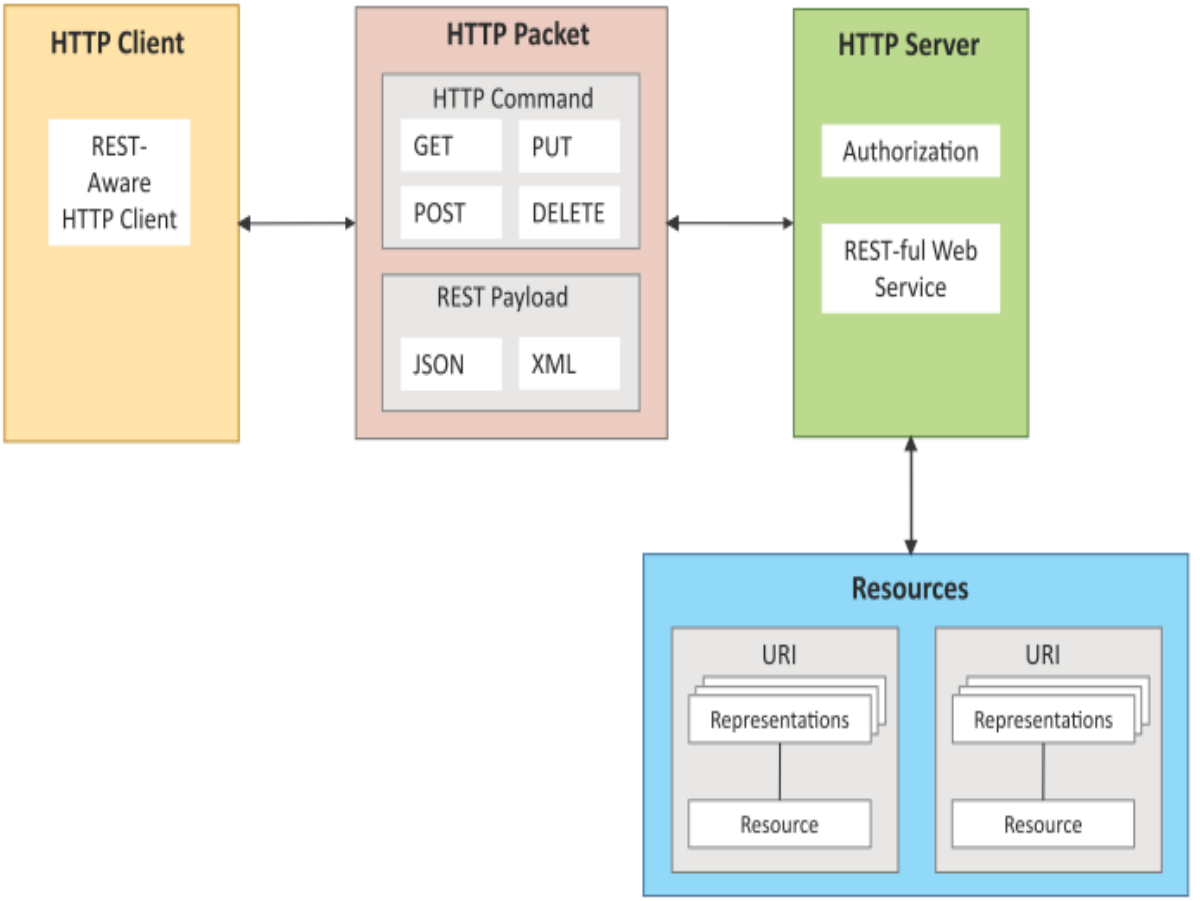
Source: Book website: http://www.internet-of-things-book.com

REST based APIs

Representational State Transfer (REST) is a set of architectural principles by which you can design web services and web APIs that focus on a system's resources and how resource states are addressed and transferred..

REST APIs follow the request–response communication model.

• REST architectural constraints apply to the components, connectors and data elements within a distributed hypermedia system



Source: Book website: http://www.internet-of-things-book.com