Chapter 1

Introduction to IoT

Outline

- IoT definition
- Characteristics of IoT
- Physical Design of IoT
- Logical Design of IoT
- IoT Protocols
- IoT Levels & Deployment Templates

Definition of IoT

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

Physical Design of IoT

- The "Things" in IoT usually refers to IoT devices which have unique identities and can perform remote sensing, actuating and monitoring capabilities.

- IoT devices can:
  - Exchange data with other connected devices and applications (directly or indirectly), or
  - Collect data from other devices and process the data locally or
  - Send the data to centralized servers or cloud-based application back-ends for processing the data, or
  - Perform some tasks locally and other tasks within the IoT infrastructure, based on temporal and space constraints
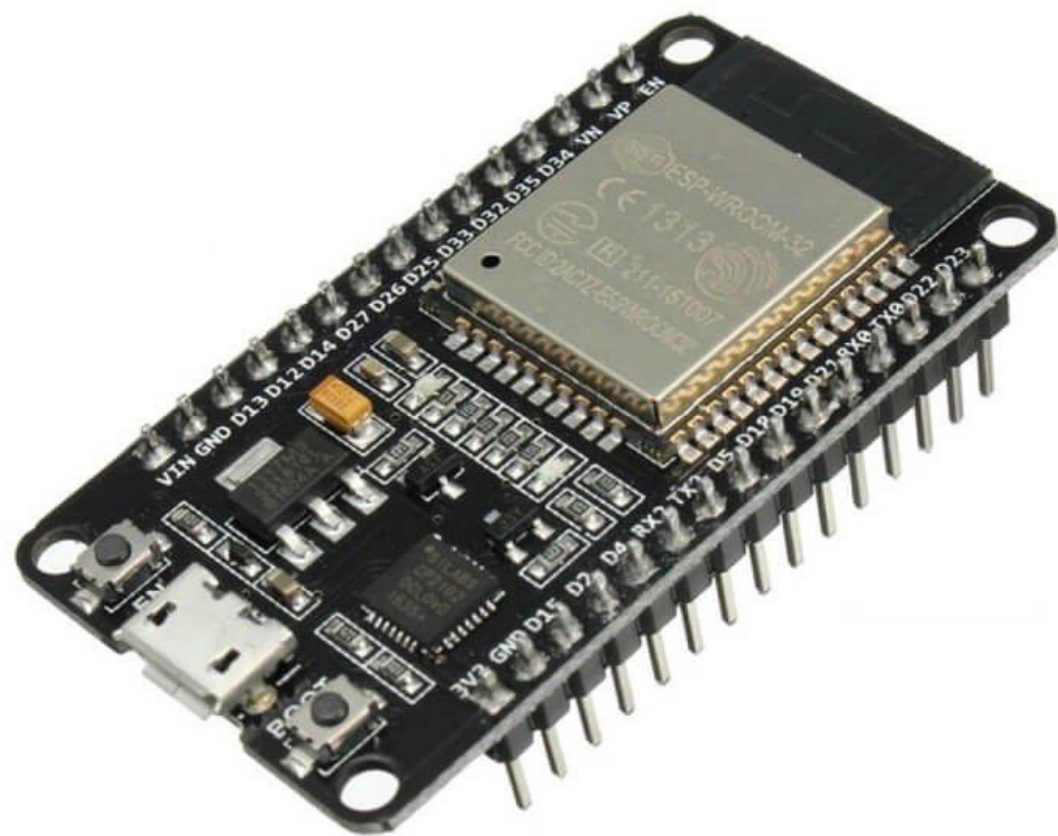
Generic block diagram of an IoT Device

- An IoT device may consist of several interfaces for connections to other devices, both wired and wireless.
  - I/O interfaces for sensors
  - Interfaces for Internet connectivity
  - Memory and storage interfaces
 - Audio/video interfaces.
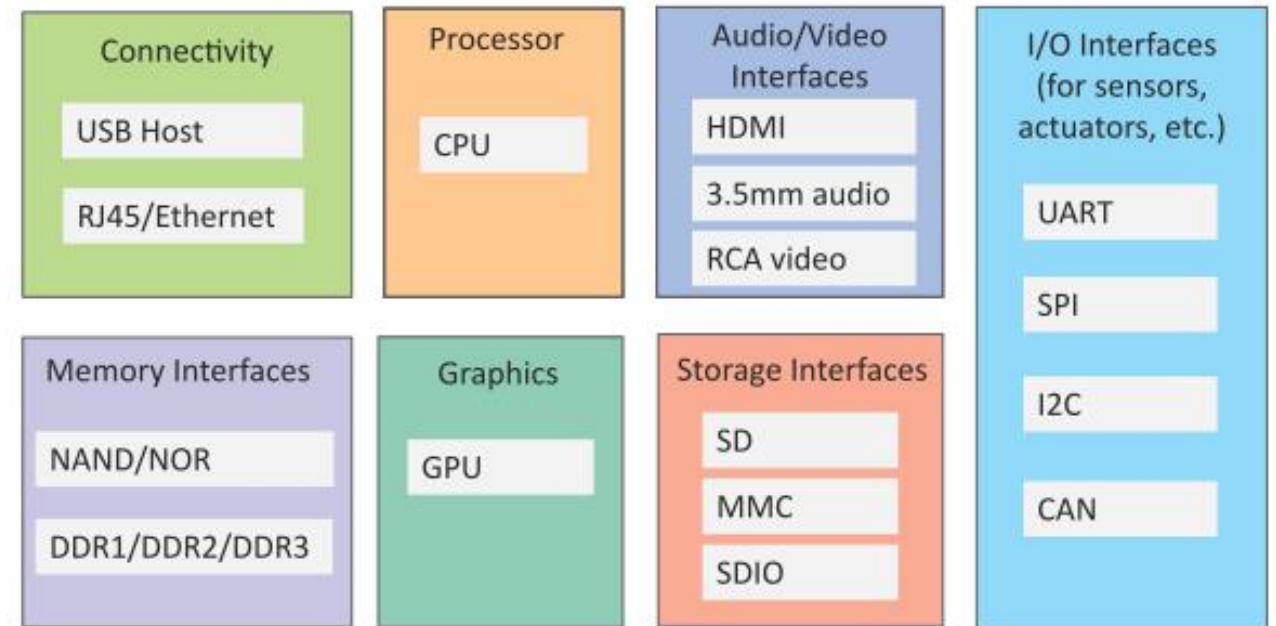
IoT Protocols

- Link Layer
  - 802.3 – Ethernet
  - 802.11 – WiFi
  - 802.16 – WiMax
  - 802.15.4 – LR-WPAN
  - 2G/3G/4G
- Network/Internet Layer
  - IPv4
  - IPv6
  - 6LoWPAN
- Transport Layer
  - TCP
  - UDP
- Application Layer
  - HTTP
  - CoAP
  - WebSocket
  - MQTT
  - XMPP
  - DDS
  - AMQP

Application Layer

HTTP	CoAP	WebSockets
MQTT	XMPP	DDS	AMQP

Transport Layer

| TCP | UDP |

Network Layer

| IPv4 | IPv6 | 6LoWPAN |

Link Layer

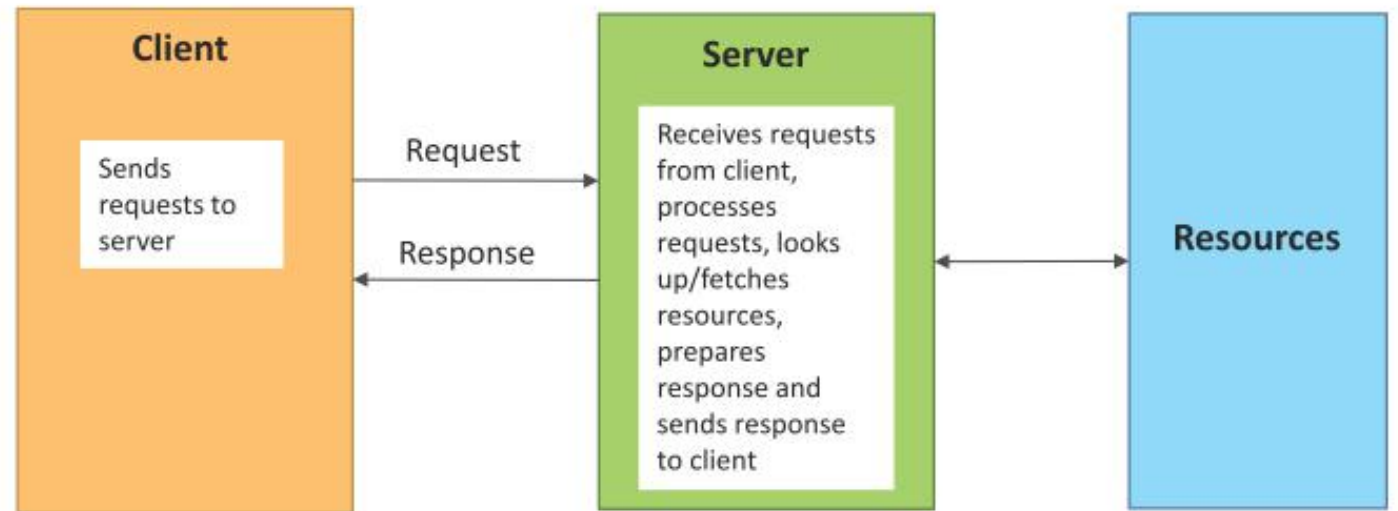| 802.3 - Ethernet | 802.16 - WiMax | 2G/3G/LTE – Cellular |
| 802.11 - WiFi | 802.15.4 – LR-WPAN | |
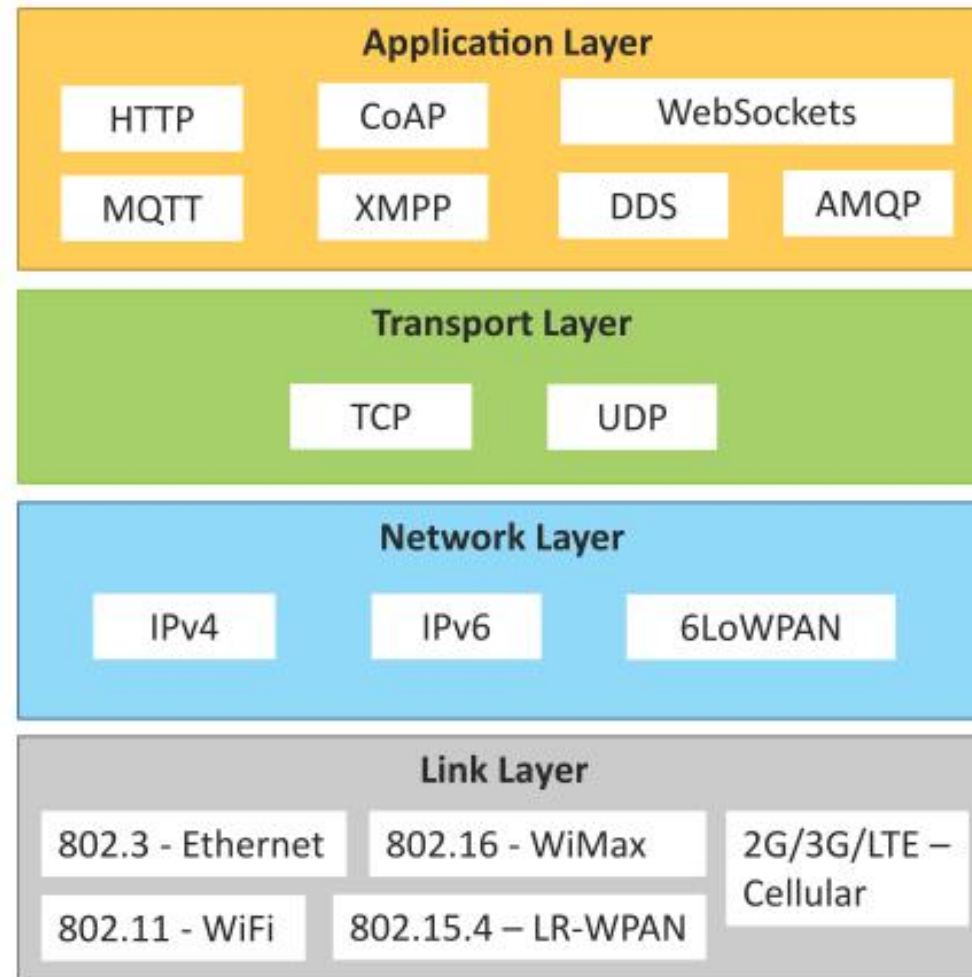
Logical Design of IoT

- 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 of a number of functional blocks that provide the system the capabilities for identification, sensing, actuation, communication, and management.
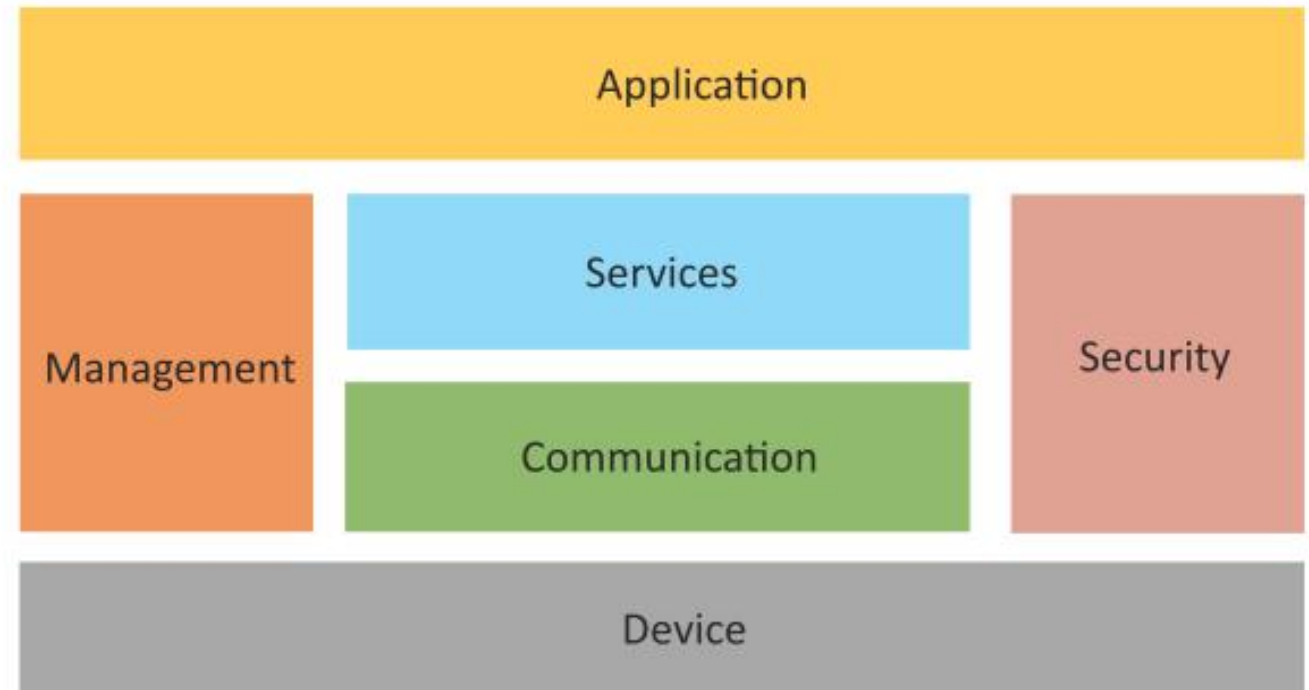
Request-Response communication model

- 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.

Client

Sends requests to server

Request

Response

Server

Receives requests from client, processes requests, looks up/fetches resources, prepares response and sends response to client

Resources

Publish-Subscribe communication model

- 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.
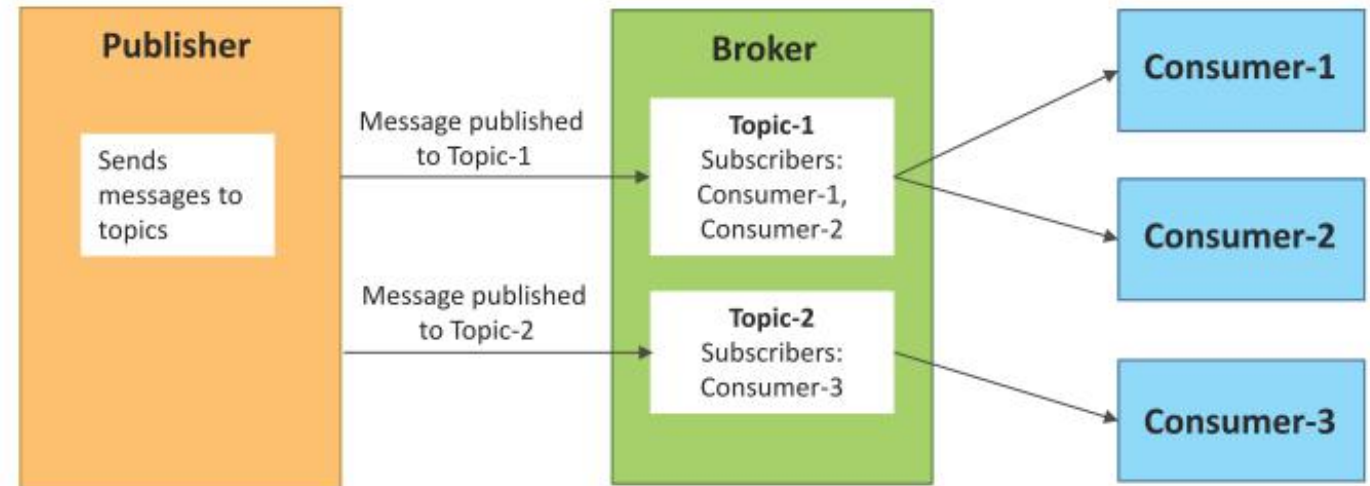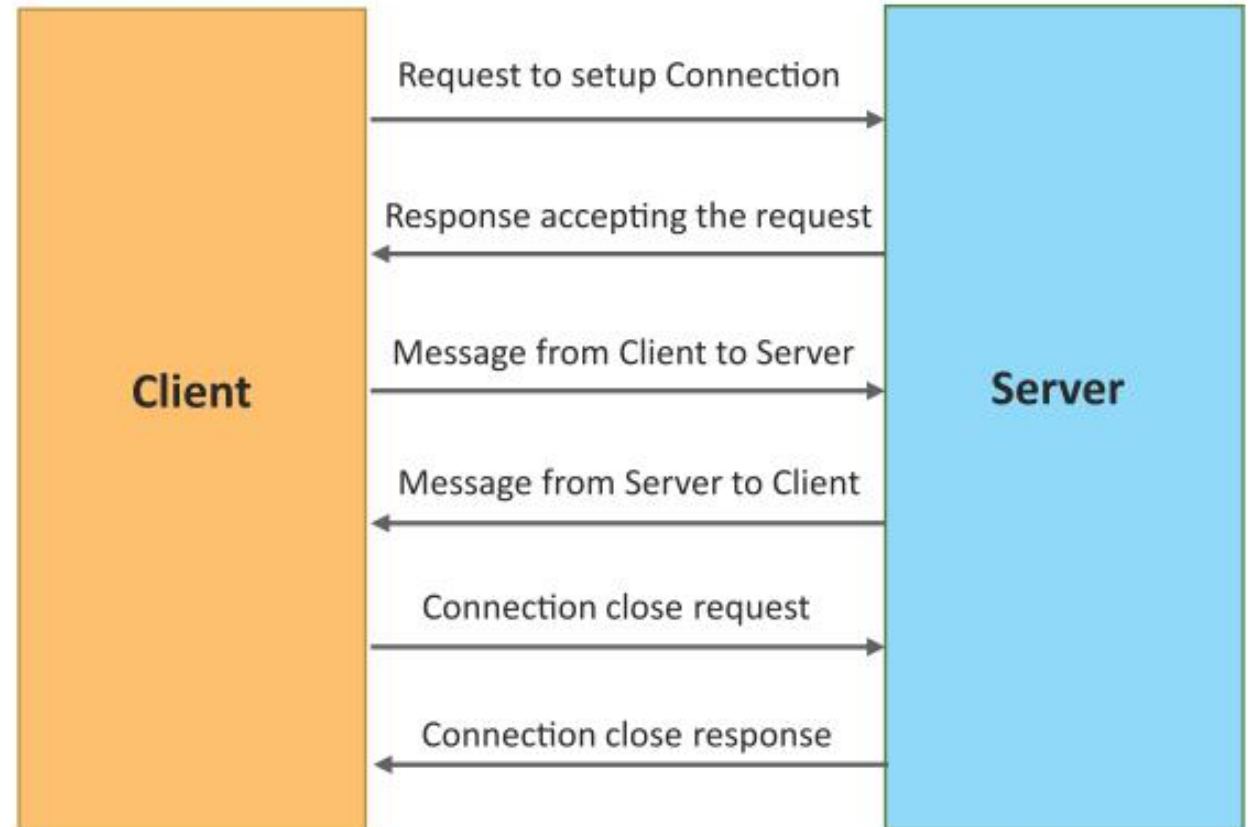
Push-Pull communication model

- 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 rate at which the consumers pull data.
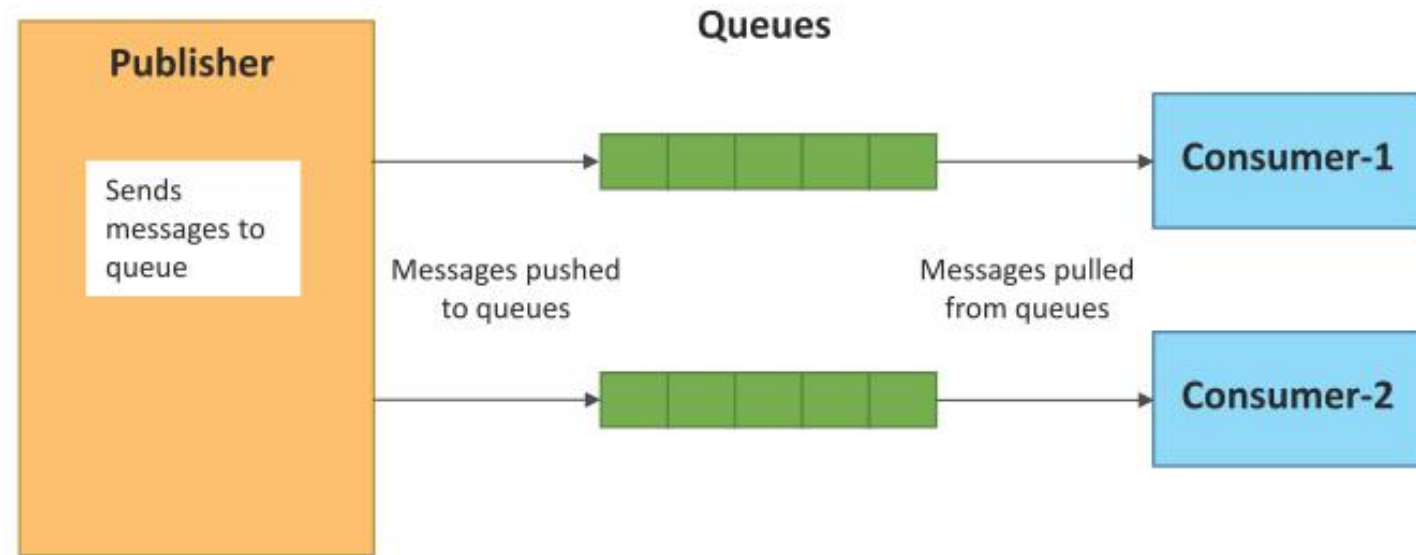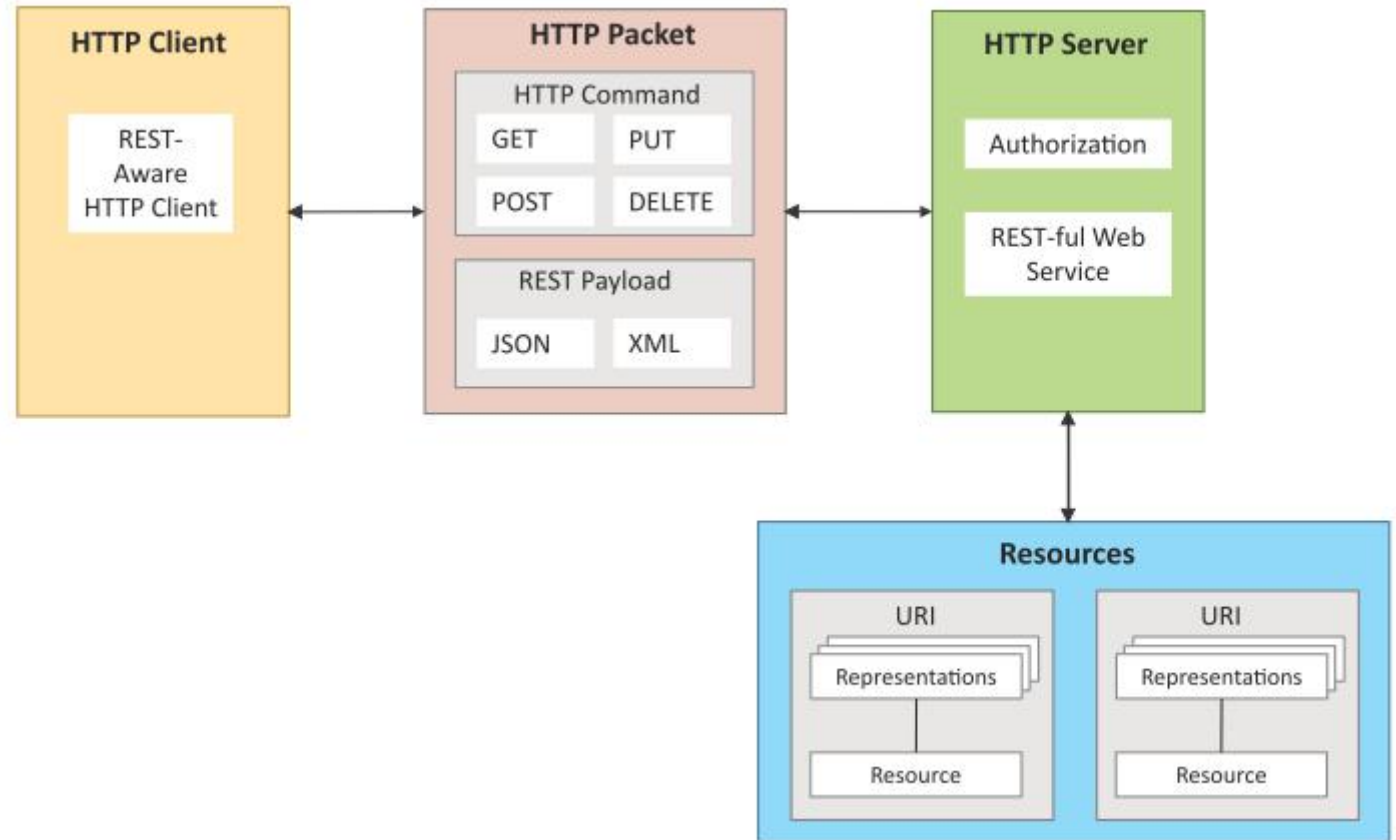
Exclusive Pair communication model

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

- Once the connection is setup it remains open until the client sends a request to close the connection.

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

Client

Request to setup Connection

Response accepting the request

Message from Client to Server

Message from Server to Client

Connection close request
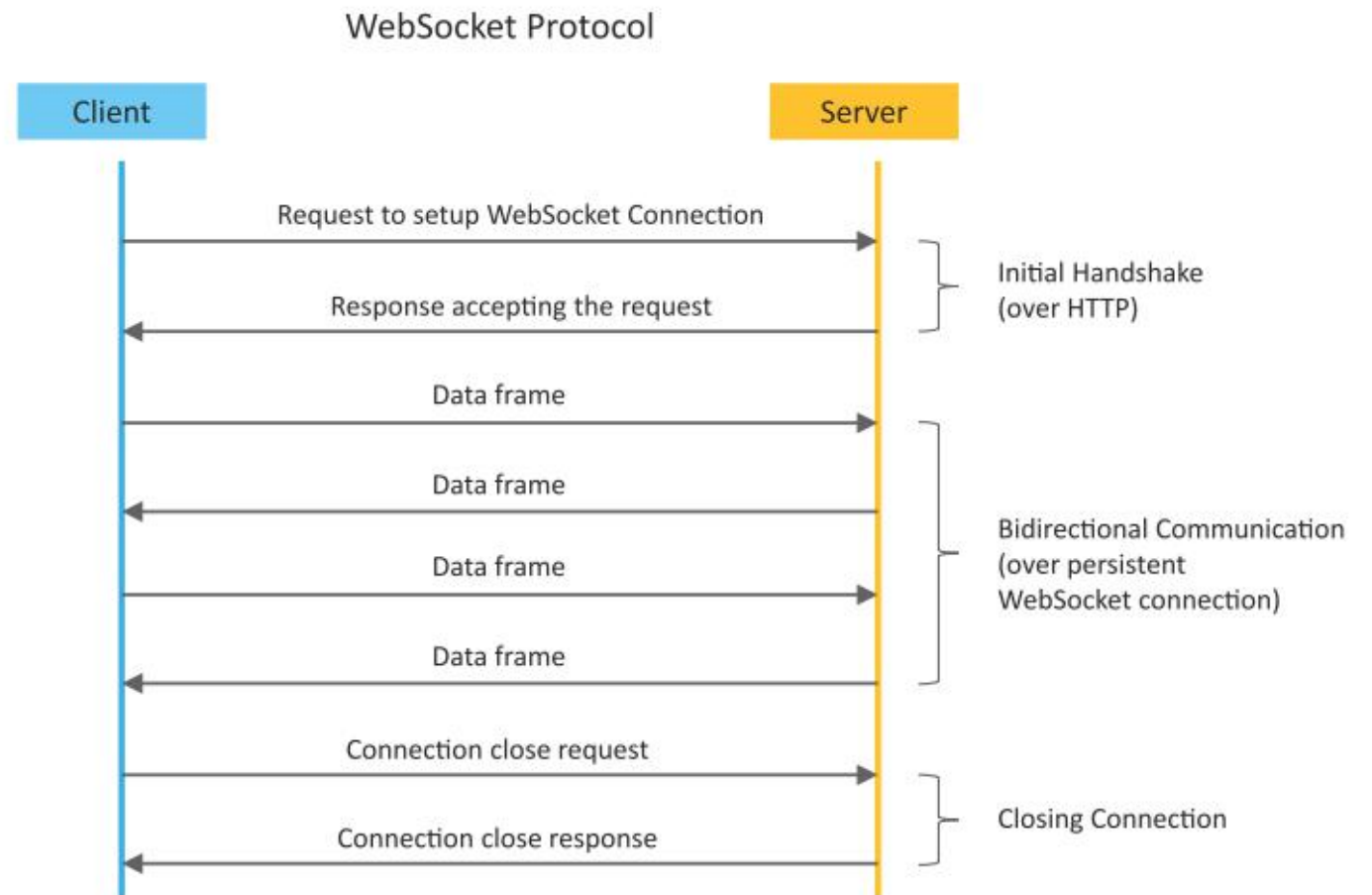
Connection close response

Server

REST-based Communication 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.

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

WebSocket-based Communication APIs

- WebSocket APIs allow bi-directional, full duplex communication between clients and servers.

- WebSocket APIs follow the exclusive pair communication model

WebSocket Protocol

Client — Server

Request to setup WebSocket Connection
Response accepting the request
Initial Handshake (over HTTP)

Data frame
Data frame
Data frame
Data frame
Bidirectional Communication (over persistent WebSocket connection)

Connection close request
Connection close response
Closing Connection

IoT Levels & Deployment Templates

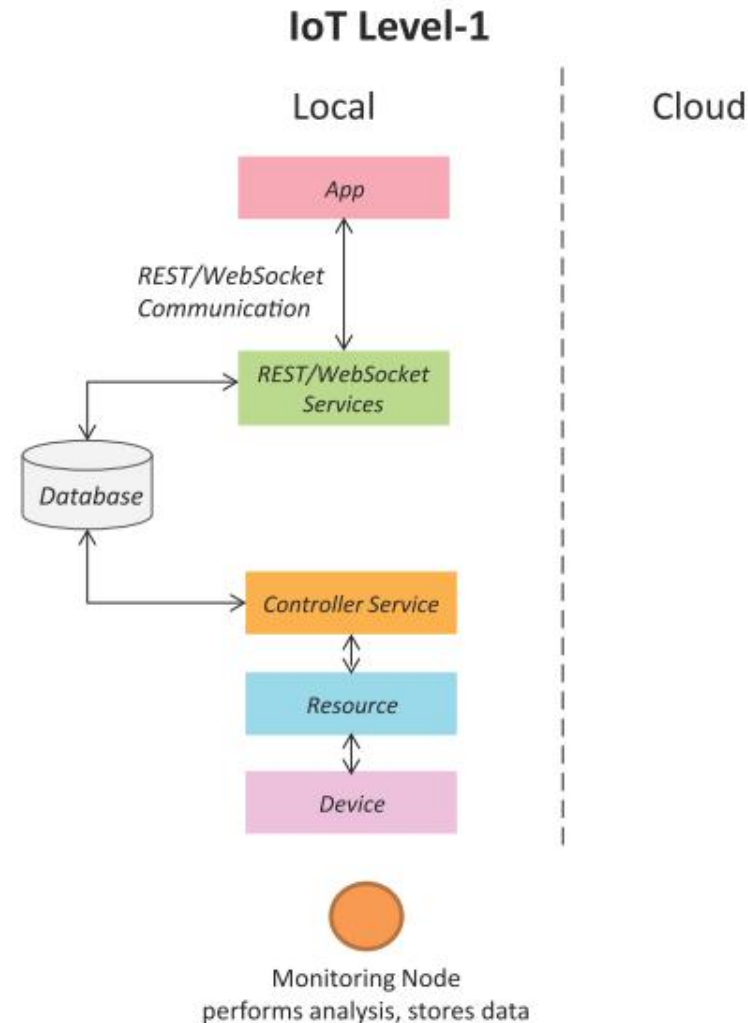An IoT system comprises of the following components:

- **Device**: An IoT device allows identification, remote sensing, actuating and remote monitoring capabilities. You learned about various examples of IoT devices in section

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

- **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 Levels & Deployment Templates

- **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 either implemented using HTTP and REST principles (REST service) or using WebSocket protocol (WebSocket service).

- **Analysis Component**: The Analysis Component is responsible for analyzing the IoT data and generate results in a form which are easy for the user to understand.

- **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 view the processed data.
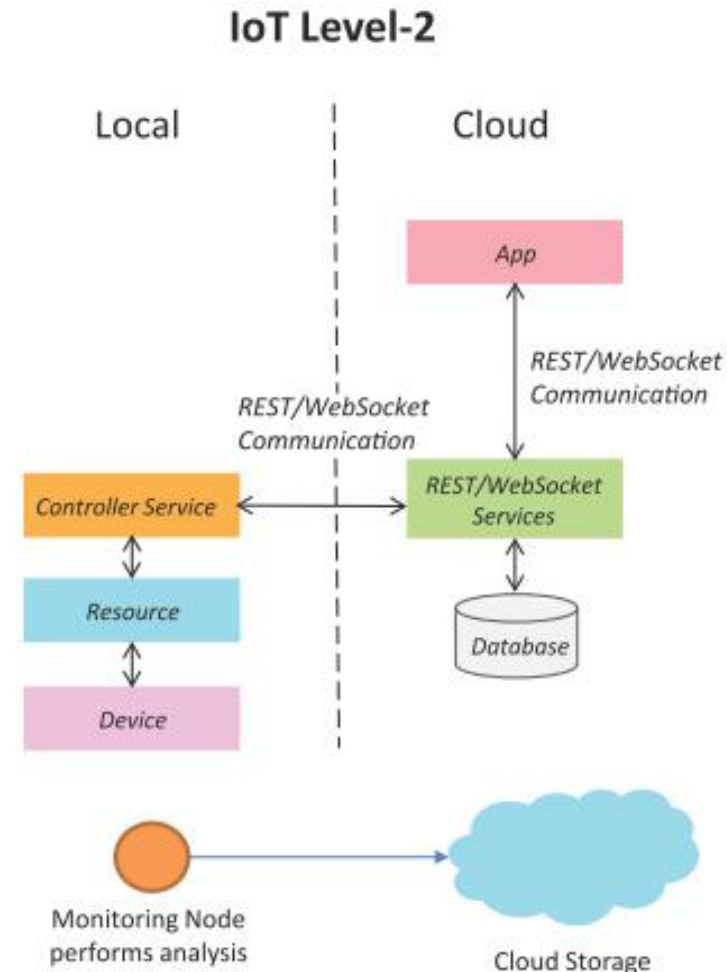
IoT Level-1

- A level-1 IoT system has a single node/device that performs sensing and/or actuation, stores data, performs analysis and hosts the application

- Level-1 IoT systems are suitable for modeling low-cost and low-complexity solutions where the data involved is not big and the analysis requirements are not computationally intensive.

IoT Level-1

Local | Cloud

App

REST/WebSocket Communication

REST/WebSocket Services

Database

Controller Service

Resource

Device

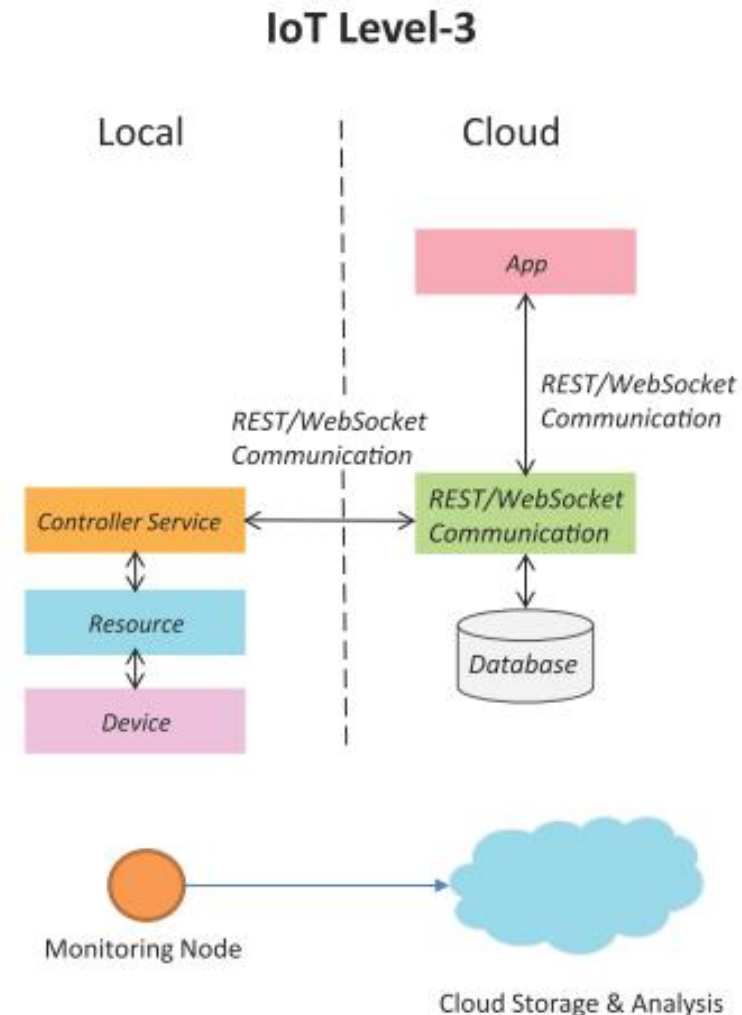Monitoring Node performs analysis, stores data

IoT Level-2

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

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

- Level-2 IoT systems are suitable for solutions where the data involved is big, however, the primary analysis requirement is not computationally intensive and can be done locally itself.
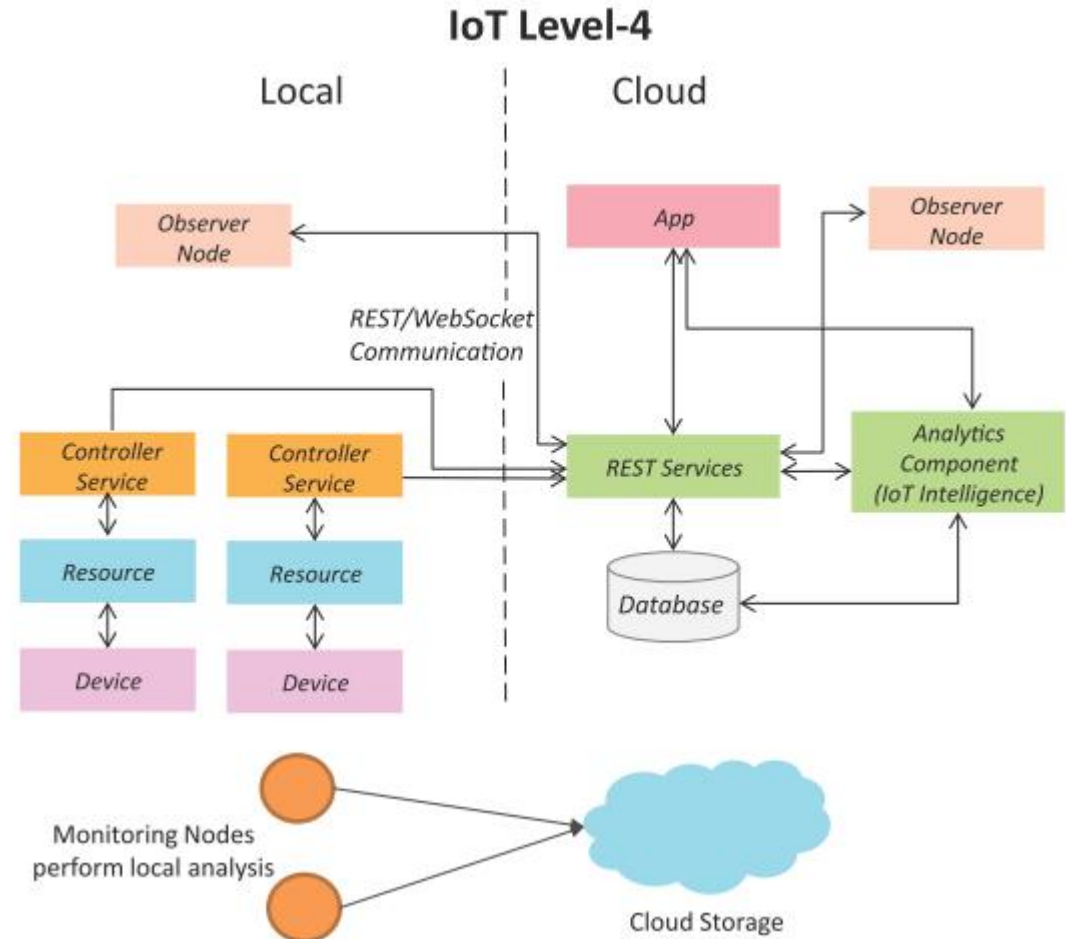
IoT Level-3

- A level-3 IoT system has a single node. Data is stored and analyzed in the cloud and application is cloud-based.

- Level-3 IoT systems are suitable for solutions where the data involved is big and the analysis requirements are computationally intensive.

IoT Level-3

Local | Cloud

App

REST/WebSocket Communication

REST/WebSocket Communication

Controller Service

REST/WebSocket Communication

Resource

Database

Device

Monitoring Node

Cloud Storage & Analysis

IoT Level-4

- A level-4 IoT system has multiple nodes that perform local analysis. Data is stored in the cloud and application is cloud-based.

- Level-4 contains local and cloud-based observer nodes which can subscribe to and receive information collected in the cloud from IoT devices.

- Level-4 IoT systems are suitable for solutions where multiple nodes are required, the data involved is big and the analysis requirements are computationally intensive.

IoT Level-5

- A level-5 IoT system has multiple end nodes and one coordinator node.

- The end nodes that perform sensing and/or actuation.

- Coordinator node collects data from the end nodes and sends to the cloud.

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

- Level-5 IoT systems are suitable for solutions based on wireless sensor networks, in which the data involved is big and the analysis requirements are computationally intensive.
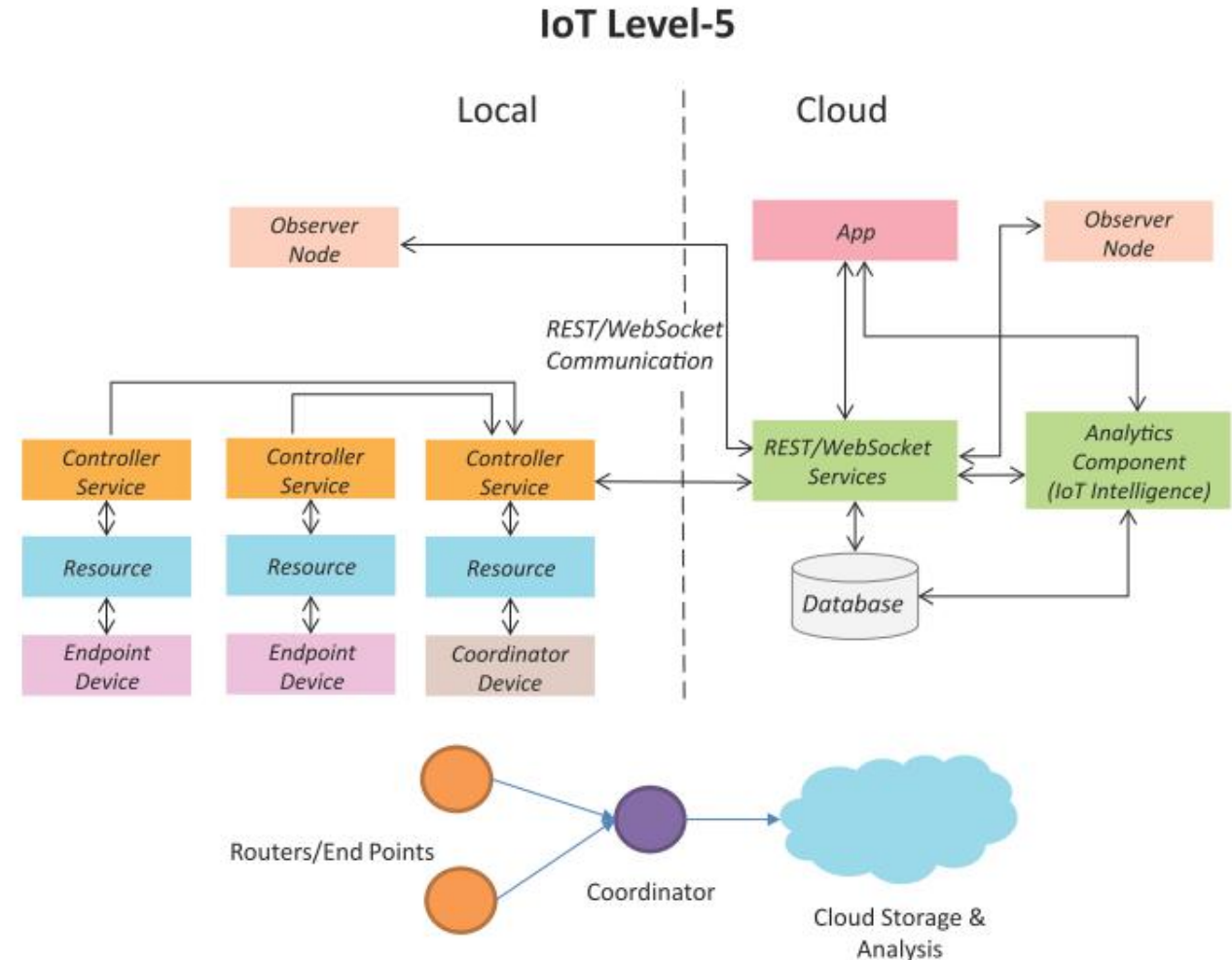
IoT Level-6

- A level-6 IoT system has 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 analyzes the data and stores the results in the cloud database.

- The results are visualized with the cloud-based application.

- The centralized controller is aware of the status of all the end nodes and sends control commands to the nodes.