Sonawane Khushbu K.

INTERNET OF THINGS(IOT)

Unit 1: Introduction to Internet of Things

- IoT stands for Internet of Things. It's a network of physical objects, such as vehicles, appliances, and other devices, that are embedded with sensors, software, and network connectivity. This allows them to collect and share data.

- **Some benefits of IoT include:**
- Real-time asset visibility
- Reduced costs
- Improved operational efficiency
- Data-driven insights for quick decision-making
- End-to-end, remote monitoring and management of assets

Internet of Things (IOT)

IoT

- Agricultural IoT
- IoT farming applications can help farmers optimize their work by sensing soil moisture and nutrients. This can help them determine when to harvest and create fertilizer profiles.

- Automated cars
- Connected cars can collect and share data with other connected devices.
-

- Infrastructure IoT
- Smart buildings can use IoT to reduce energy consumption, maintain costs, and use space more efficiently.

- Smart cities
- Smart cities can improve traffic flow, reduce waste, and increase security.

- IoT applications in logistics
- IoT can help logistic providers with operational efficiencies. This can include projects such as inventory management, storage management, and condition monitoring.
-

Characteristics of IoT

- **1. Connectivity**
- Connectivity is an important requirement of the IoT infrastructure. Things of IoT should be connected to the IoT infrastructure. Anyone, anywhere, anytime can connect, this should be guaranteed at all times. For example, the connection between people through Internet devices like mobile phones, and other gadgets, also a connection between Internet devices such as routers, gateways, sensors, etc.

- **2. Intelligence and Identity**
- The extraction of knowledge from the generated data is very important. For example, a sensor generates data, but that data will only be useful if it is interpreted properly. Each IoT device has a unique identity. This identification is helpful in tracking the equipment and at times for querying its status.

- **3. Scalability**
- The number of elements connected to the IoT zone is increasing day by day. Hence, an IoT setup should be capable of handling the massive expansion. The data generated as an outcome is enormous, and it should be handled appropriately.

- **4. Dynamic and Self-Adapting (Complexity)**
- IoT devices should dynamically adapt themselves to changing contexts and scenarios. Assume a camera meant for surveillance. It should be adaptable to work in different conditions and different light situations (morning, afternoon, and night).

- **5. Architecture**
- IoT Architecture cannot be homogeneous in nature. It should be hybrid, supporting different manufacturers ' products to function in the IoT network. IoT is not owned by anyone engineering branch. IoT is a reality when multiple domains come together.

- **6. Safety**
- There is a danger of the sensitive personal details of the users getting compromised when all his/her devices are connected to the internet. This can cause a loss to the user. Hence, data security is the major challenge. Besides, the equipment involved is huge. IoT networks may also be at risk. Therefore, equipment safety is also critical.

- **8. Interoperability**
- IoT devices use standardized protocols and technologies to ensure they can communicate with each other and other systems. Interoperability is one of the key characteristics of the Internet of Things (IoT). It refers to the ability of different IoT devices and systems to communicate and exchange data with each other, regardless of the underlying technology or manufacturer.
- heart rate monitor (HRM)

- Interoperability is critical for the success of IoT, as it enables different devices and systems to work together seamlessly and provides a seamless user experience. Without interoperability, IoT systems would be limited to individual silos of data and devices, making it difficult to share information and create new services and applications.

- To achieve interoperability, IoT devices, and systems use standardized communication protocols and data formats. These standards allow different devices to understand and process data in a consistent and reliable manner, enabling data to be exchanged between devices and systems regardless of the technology used.

- **Examples of standards used in IoT**
- **MQTT (Message Queuing Telemetry Transport):** MQTT (Message Queuing Telemetry Transport) is a publish/subscribe communication protocol used for IoT device communication.
- **CoAP (Constrained Application Protocol):** CoAP (Constrained Application Protocol) is a lightweight communication protocol for IoT devices with limited resources.
- **Bluetooth Low Energy (BLE):** Bluetooth Low Energy is a wireless communication technology used for IoT devices with low power consumption requirements.

- **Wi-Fi:** A wireless communication technology used for IoT devices that require high data transfer rates.
- **Zigbee:** A low-power, low-cost wireless communication technology used for IoT devices.
- In addition to communication protocols, IoT systems may also use data formats such as JSON or XML to ensure that data can be exchanged and processed consistently across different systems.
- Overall, interoperability is essential for creating a seamless IoT ecosystem, where devices and systems can work together to deliver new and innovative services and applications.

- **9. Embedded Sensors and Actuators**

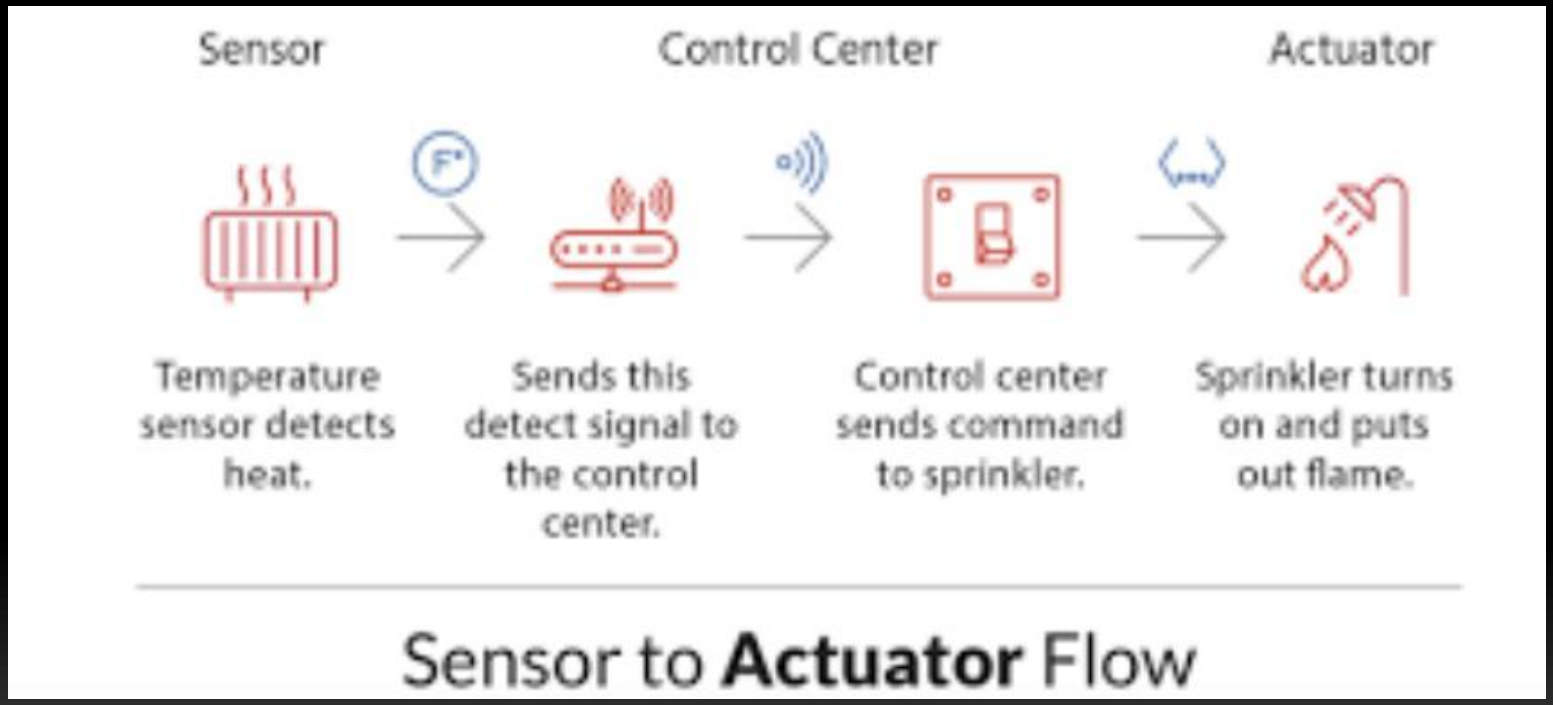
- Embedded sensors and actuators are critical components of the Internet of Things (IoT). They allow IoT devices to interact with their environment and collect and transmit data.

- Sensors are devices that can detect changes in the environment, such as temperature, light, sound, or movement. In IoT systems, sensors are embedded into devices, allowing them to collect data about the environment.

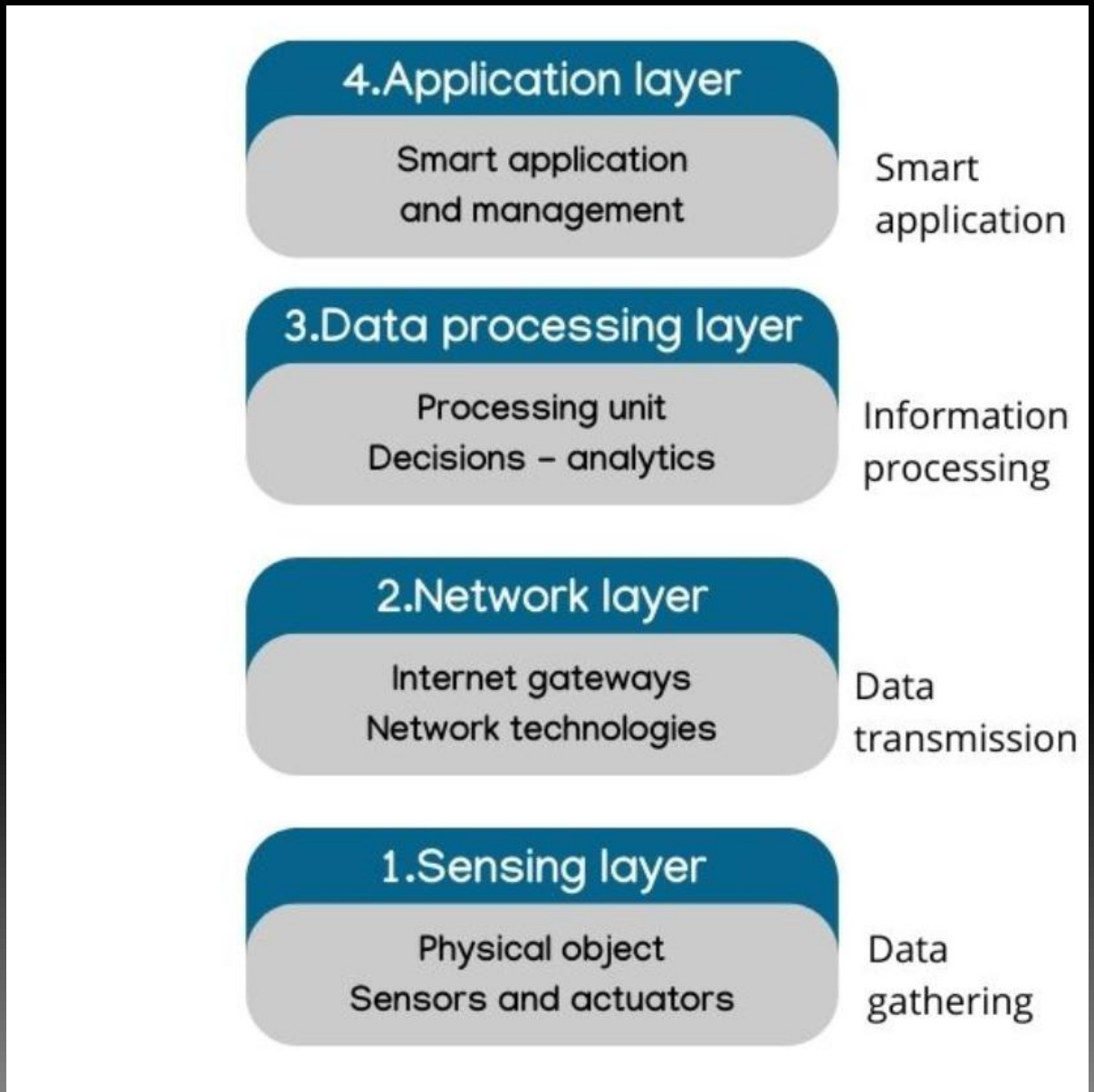
- Actuators are devices that can interact with the environment, such as turning on lights, opening or closing doors, or controlling the speed of a motor. In IoT systems, actuators are embedded into devices, allowing them to perform actions based on data collected by sensors.
- Together, sensors and actuators allow IoT devices to collect data about the environment, process that data, and take action based on the results. This makes it possible to automate a wide range of processes and tasks, such as home automation, energy management, and predictive maintenance.

- In order to ensure that sensors and actuators can communicate with each other and with other devices and systems, they use standardized communication protocols, such as Bluetooth Low Energy (BLE), Zigbee, or Wi-Fi.
- Overall, embedded sensors and actuators are essential components of IoT systems, enabling them to collect and process data and interact with their environment in new and innovative ways.
- IoT devices are equipped with sensors and actuators that allow them to collect and transmit data, as well as to interact with the environment.

- Actuators in the Internet of Things (IoT) are devices that convert energy into motion. They are essential for the operation of IoT devices because they enable the automation and control of physical systems and machinery


Sensor
Control Center
Actuator
Temperature sensor detects heat.
Sends this detect signal to the control center.
Control center sends command to sprinkler.
Sprinkler turns on and puts out flame.
Sensor to Actuator Flow

Understanding of IoT Architecture



- **Sensing Layer –**
  The sensing layer is the first layer of the IoT architecture and is responsible for collecting data from different sources. This layer includes sensors and actuators that are placed in the environment to gather information about temperature, humidity, light, sound, and other physical parameters. These devices are connected to the network layer through wired or wireless communication protocols.

- **Network Layer –**
  The network layer of an IoT architecture is responsible for providing communication and connectivity between devices in the IoT system. It includes protocols and technologies that enable devices to connect and communicate with each other and with the wider internet. Examples of network technologies that are commonly used in IoT include WiFi, Bluetooth, Zigbee, and cellular networks such as 4G and 5G. Additionally, the network layer may include gateways and routers that act as intermediaries between devices and the wider internet, and may also include security features such as encryption and authentication to protect against unauthorized access.

- **Data processing Layer –**
  The data processing layer of IoT architecture refers to the software and hardware components that are responsible for collecting, analyzing, and interpreting data from IoT devices. This layer is responsible for receiving raw data from the devices, processing it, and making it available for further analysis or action.The data processing layer includes a variety of technologies and tools, such as data management systems, analytics platforms, and machine learning algorithms. These tools are used to extract meaningful insights from the data and make decisions based on that data.**Example** of a technology used in the data processing layer is a data lake, which is a centralized repository for storing raw data from IoT devices.

- **Application Layer –**
  The application layer of IoT architecture is the topmost layer that interacts directly with the end-user. It is responsible for providing user-friendly interfaces and functionalities that enable users to access and control IoT devices.This layer includes various software and applications such as mobile apps, web portals, and other user interfaces that are designed to interact with the underlying IoT infrastructure. It also includes middleware services that allow different IoT devices and systems to communicate and share data seamlessly.The application layer also includes analytics and processing capabilities that allow data to be analyzed and transformed into meaningful insights. This can include machine learning algorithms, data visualization tools, and other advanced analytics capabilities.

Smart Lighting

- Smart lighting for homes helps in saving energy by adapting the lighting to the ambient condition and switching on/off or dimming the lights when needed.

- Smart lighting solution for home achieve energy saving by sensing the human movements and their environments and controlling the lights accordingly.

Smart Appliances

- Smart appliances make the management easier and also provide status information to the users remotely.
- Example: smart washer/dryer can be controlled remotely and notify when the washing/ drying is complete.
- Smart refrigerators can keep track of the store and send updates to the users when an items is low on stalk.

Intrusion Detection

- network intrusion is any illegal activity carried out on a digital network
- Home intrusion detection system users security cameras and sensors to detect intrusion and raise alerts.
- Alerts can be in the form of sms or an email sent to the user.
- Advanced system can even send detailed alerts such as an image grab or short video clip.

Smoke/Gas Detectors

- Smoke detectors are installed in home and buildings to detect smoke that is typically n early sign of fire.
- It users optical detection, ionization or an sampling techniques to detect smoke.
- Gas detector can detect the presence of harmful gases such as CO, LPG etc.
- It can raise alerts in human voice describing where the problem is.

Smart Parking

- It make the search for parking space easier and convenient for dinners.
- These are powered by IOT systems that detect the no of empty parking slots and send the information over the internet to smart parking application back-ends.

Smart Lighting

- It allows lighting to be dynamically controlled remotely to configure lighting schedules and lighting intensity.
- Customer lighting configuration can be set of for different situations such as a foggy day, a festival etc.
- Smart light are equipped with sensors that can communicate with other lights and exchange information on the sensed ambient condition to adopt the lighting.

Smart Roads

- Smart roads can provide info on driving conditions, travel time estimates and alerts in case of poor driving condition, traffic congestions and accidents.

- Such info can help in making the roads safer and help in reducing traffic jams.

Structural Health Monitoring

- This system users a network of sensors to monitor the vibration levels in the structures such as bridges and buildings.
- The data collected from these sensors is analyzed to access the health of the structures detecting cracks and mechanical breakdown, remaining life of the structure.

Surveillance

- Surveillance of infrastructure public transport and events in cities is required to ensure safety and security.
- City wide surveillance infrastructure comprising to large number of distributed and internet connected video surveillance cameras can be created.

Emergency Response

- IOT systems can be used for monitoring the critical infrastructure in cities such as building gas and water pipelines, public transport and power stations.
- Fire detection, gas and water leakage detection can help in generating alerts and minimizing their effects on the critical infrastructure
- Such system can reduce the latency of emergency services for vehicles such as ambulances, and police cars while minimizing disruption of regular traffic.

Physical Design of IoT

- The physical design of an Internet of Things (IoT) system involves the hardware components and their arrangement to enable seamless communication and data exchange between devices. Here are key considerations for the physical design of IoT:

- **Sensors and Actuators:**
  - An actuator is a machine component or system that moves or controls the mechanism of the system
  - Identify the types of sensors and actuators needed for your specific IoT application.
  - Choose sensors that are compatible with the environmental conditions and requirements of the deployment location.

- **Embedded Systems:**
  - Utilize embedded systems to integrate sensors, processors, and communication modules into a compact and efficient package.
  - Choose microcontrollers or microprocessors that meet the processing requirements of your application.

- **Power Supply:**
  - Select an appropriate power source or combination of sources (e.g., batteries, solar panels, energy harvesting) based on the deployment environment and device power requirements.
  - Implement power-efficient designs to extend the operational life of IoT devices.

- **Communication Modules:**
  - Choose suitable communication protocols (e.g., MQTT, CoAP, HTTP) and modules (e.g., Wi-Fi, Bluetooth, Zigbee) for data transfer based on the application requirements.
  - Consider the range, data rate, and power consumption of communication modules.

- **Enclosures and Housing:**
  - Design protective enclosures to shield IoT devices from environmental factors such as moisture, dust, and extreme temperatures.
  - Consider the material and form factor of the enclosure to ensure durability and ease of installation.

- **Scalability:**
  - Design IoT devices with scalability in mind to accommodate future updates, expansions, or additional features.
  - Ensure that the physical design allows for easy integration with other devices and systems.

- **Security:**
  - Implement security measures at the physical level, such as tamper-resistant enclosures and secure boot processes.
  - Consider the physical security of the device to prevent unauthorized access or tampering.

- **Durability and Reliability:**
  - Design IoT devices to withstand the expected operational conditions and handle potential shocks, vibrations, and other stress factors.
  - Conduct thorough testing to ensure reliability over time.

- **Environmental Impact:**
  - Consider the environmental impact of IoT devices, including materials used in manufacturing and disposal considerations.
  - Aim for energy-efficient designs to minimize the ecological footprint.

- **Regulatory Compliance:**
  - Ensure that the physical design complies with relevant regulations and standards applicable to IoT devices in the target market.

- **Maintenance and Upgradability:**
  - Plan for ease of maintenance and future upgrades by designing devices with modular components and firmware update capabilities.

- By carefully considering these factors, you can create a robust and efficient physical design for your IoT devices, facilitating their integration into a larger IoT ecosystem.

IoT devices

- Internet of Things (IoT) devices are hardware devices that connect to the internet or a local network hub wirelessly. They can collect and exchange data over the internet and other networks.

- **IoT devices include:**
- Sensors, Gadgets, Appliances, Actuators, Software, Computer devices.

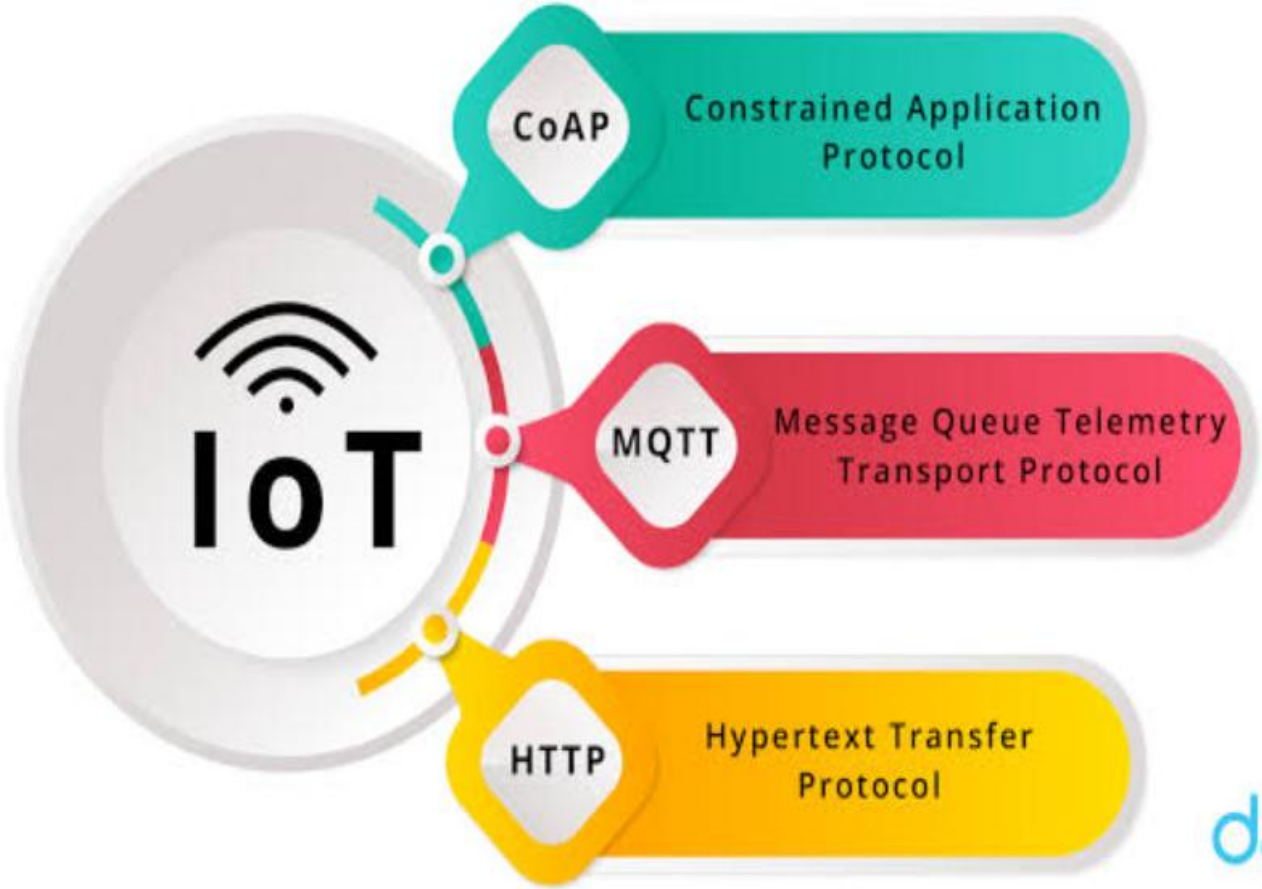
- IoT devices are programmed for certain applications and can be embedded into other IoT devices. They can also be attached to a particular object that operates through the internet. This enables the transfer of data among objects or people automatically without human intervention.
- Some examples of IoT devices include: Cars, Doorbells, Refrigerators.

- **Before buying an IoT device, you can consider things like:**
- Location: Whether the device will work where you want it to
- Data collection: What data is collected and shared, how long it's retained, and whether you can opt out
- Security: What security measures the device takes
- Access: Whether you can access the data the device collects

IoT protocols

- IoT protocols are sets of rules and standards that govern how IoT devices communicate with each other and other systems over the internet. These protocols define how data is exchanged, what format it should be in, and how it should be encrypted and authenticated for security purposes.

IoT Protocols



daviteq

- **LoRaWAN**
- A media access control (MAC) IoT protocol that allows low-powered devices to communicate directly with internet-connected applications over a long-range wireless connection. It uses very little power, has good coverage, and works well indoors.

- **MQTT**
- A standards-based messaging protocol, or set of rules, used for machine-to-machine communication. It is designed as an extremely lightweight publish/subscribe messaging transport that is ideal for connecting remote devices with a small code footprint and minimal network bandwidth.

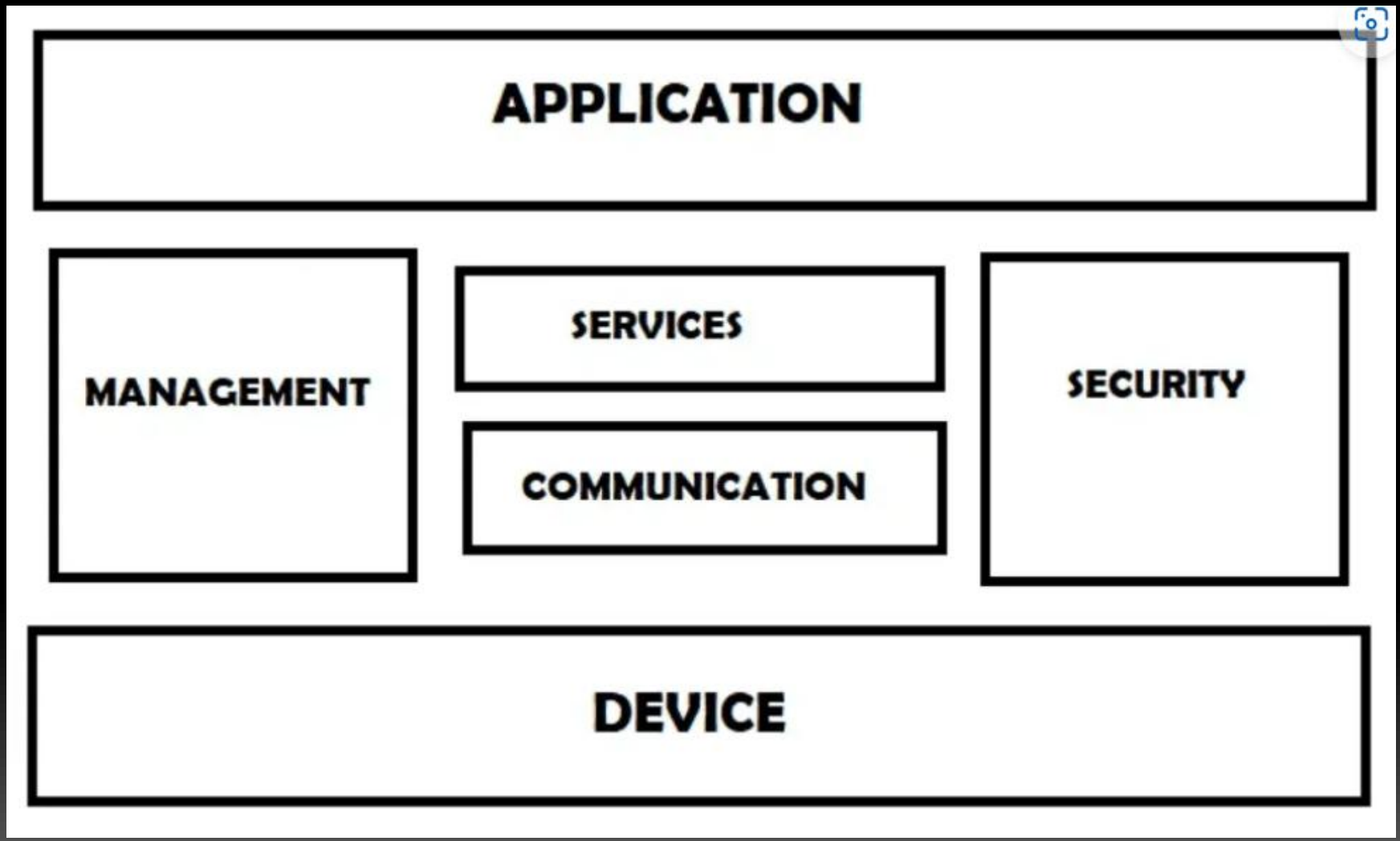
- **Bluetooth**
- A wireless communication technology that allows devices to connect and exchange data over short distances. It is a standard IoT protocol for wireless data transmission.

- **Wi-Fi**
- A local area network used for connecting a broad network of computers. It has longer range as compared to Bluetooth, Zigbee, and other protocols.

- **Constrained Application Protocol (CoAP)**
- A specialized web transfer protocol for use with constrained nodes and constrained networks in the Internet of Things.

Logical Design of IoT

- A logical design for an IoT system is the actual design of how its components (computers, sensors, and actuators) should be arranged to complete a particular function. It doesn't go into the depth of describing how each component will be built with low-level programming specifics.


APPLICATION
MANAGEMENT
SERVICES
COMMUNICATION
SECURITY
DEVICE

- The logical design of an IoT system includes: Computers, Sensors, Actuators.
- The IoT design approach covers all components of IoT architecture, from IoT devices and their hardware to applications and user interfaces.
- **The basic IoT architecture consists of three layers:**
- Perception: The sensors, gadgets, and other devices
- Network: The connectivity between devices
- Application: The layer the user interacts
- The environment itself is essential during IoT design. For example, you may want to reduce user distraction or build gadgets that are weather resistant.

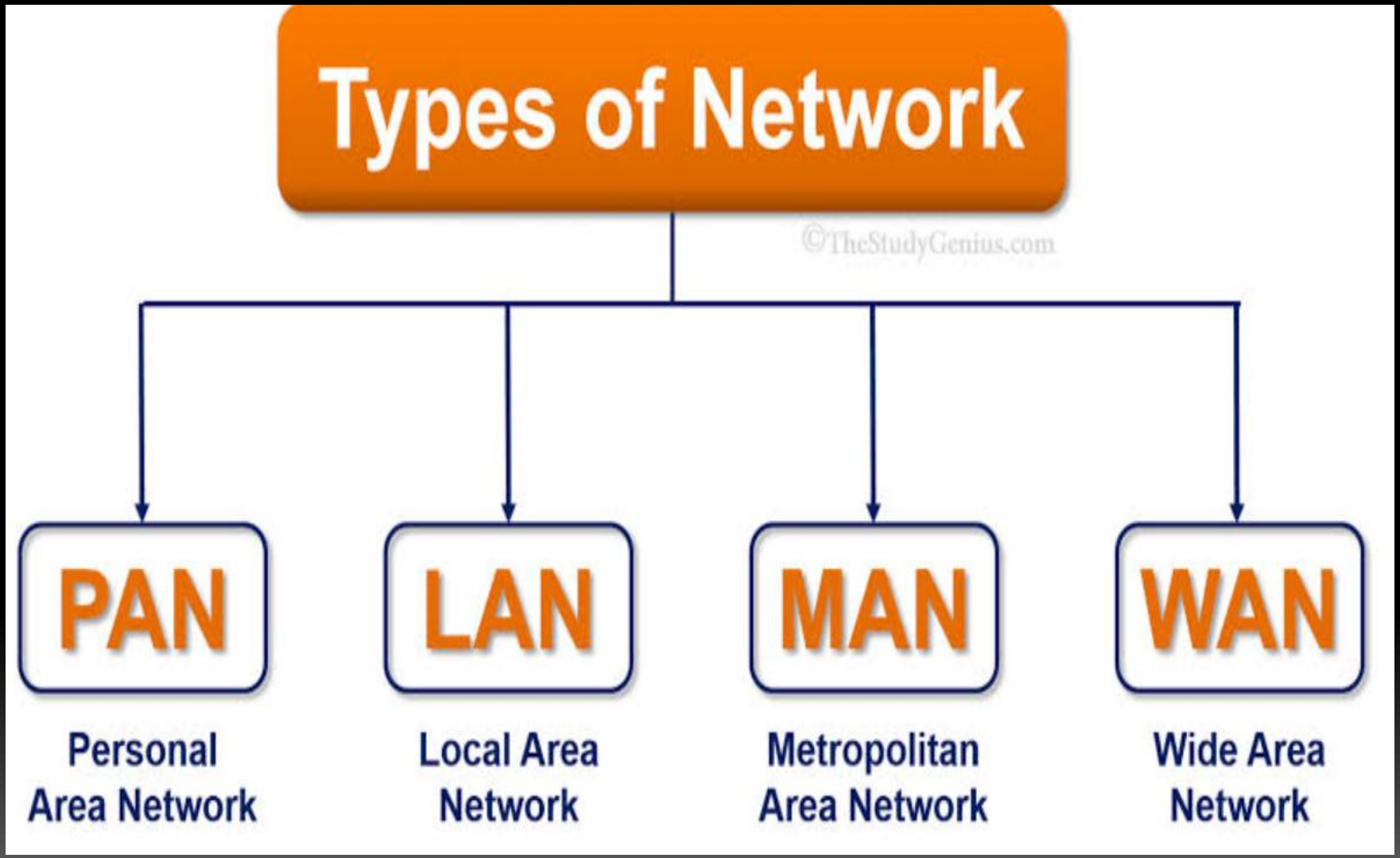
IoT Functional Blocks

- The functional blocks of Internet of Things (IoT) devices vary depending on their complexity and purpose. Some common functional blocks include:
-

- Sensors
- IoT sensors include temperature, pressure, motion, level, image, proximity, water quality, and chemical sensors

- Actuators
- Actuators are devices that convert energy into motion. They can act on data collected by sensors to create an outcome based on the user's chosen settings

- Gateways
- Gateways route processed data and transfer it to proper databases or network storage. Examples of gateways include LAN, WAN, and PAN


Types of Network
©TheStudyGenius.com
PAN
LAN
MAN
WAN
Personal Area Network
Local Area Network
Metropolitan Area Network
Wide Area Network

- Data processing and analytics
- **Other functional blocks include:**
- Processors, Connectivity modules, Power supply, Memory and storage, User interface, Security.

- **IoT systems also include four basic building blocks:**
- Sensors
- Processors
- Gateways
- Applications

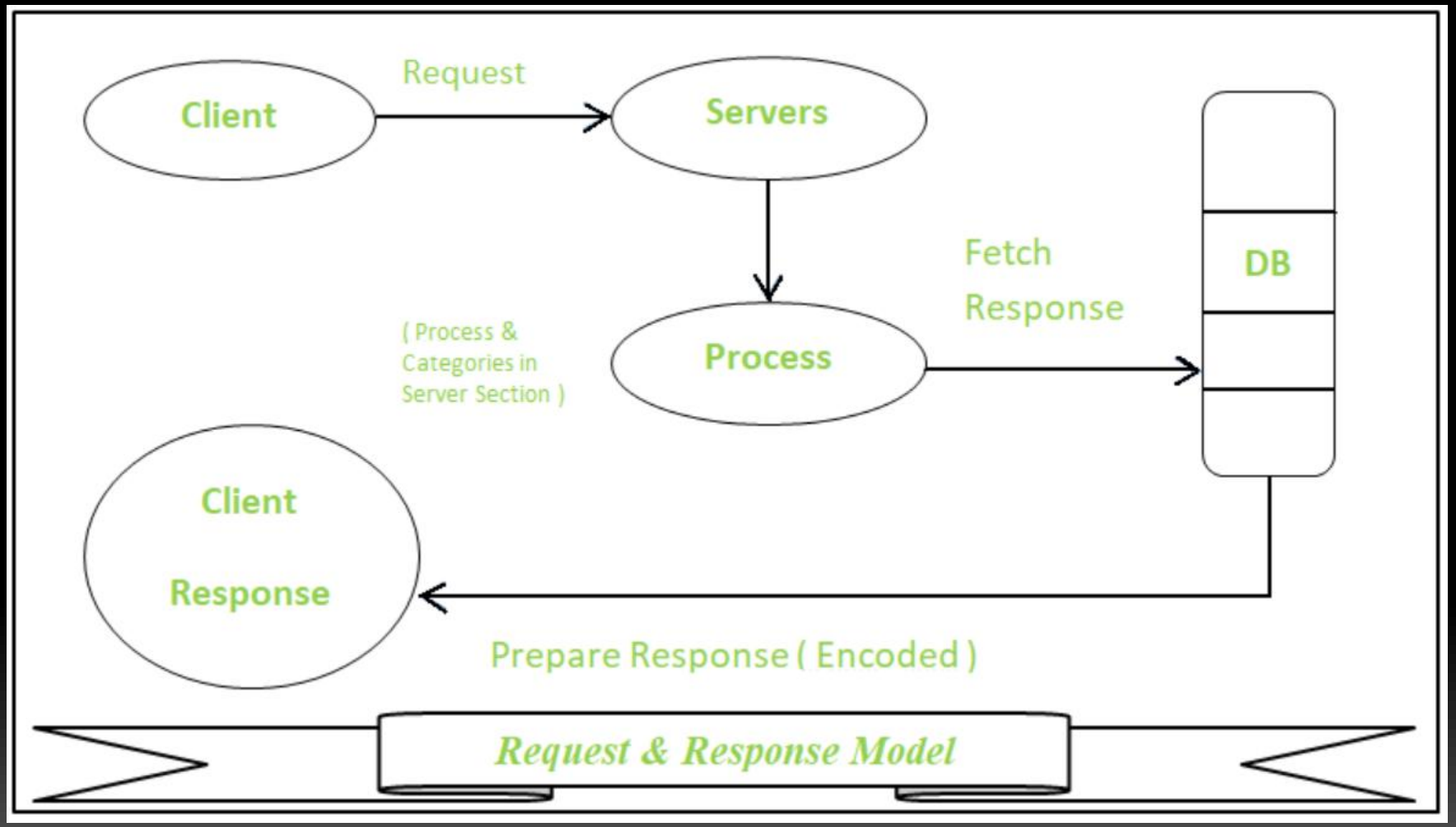
- When choosing hardware for an IoT project, you can consider things like:

- Security requirements
- Ease of development
- Data acquisition, processing, and storage requirements
- Connectivity requirements
- Power requirements
- Physical device design
- Cost requirements

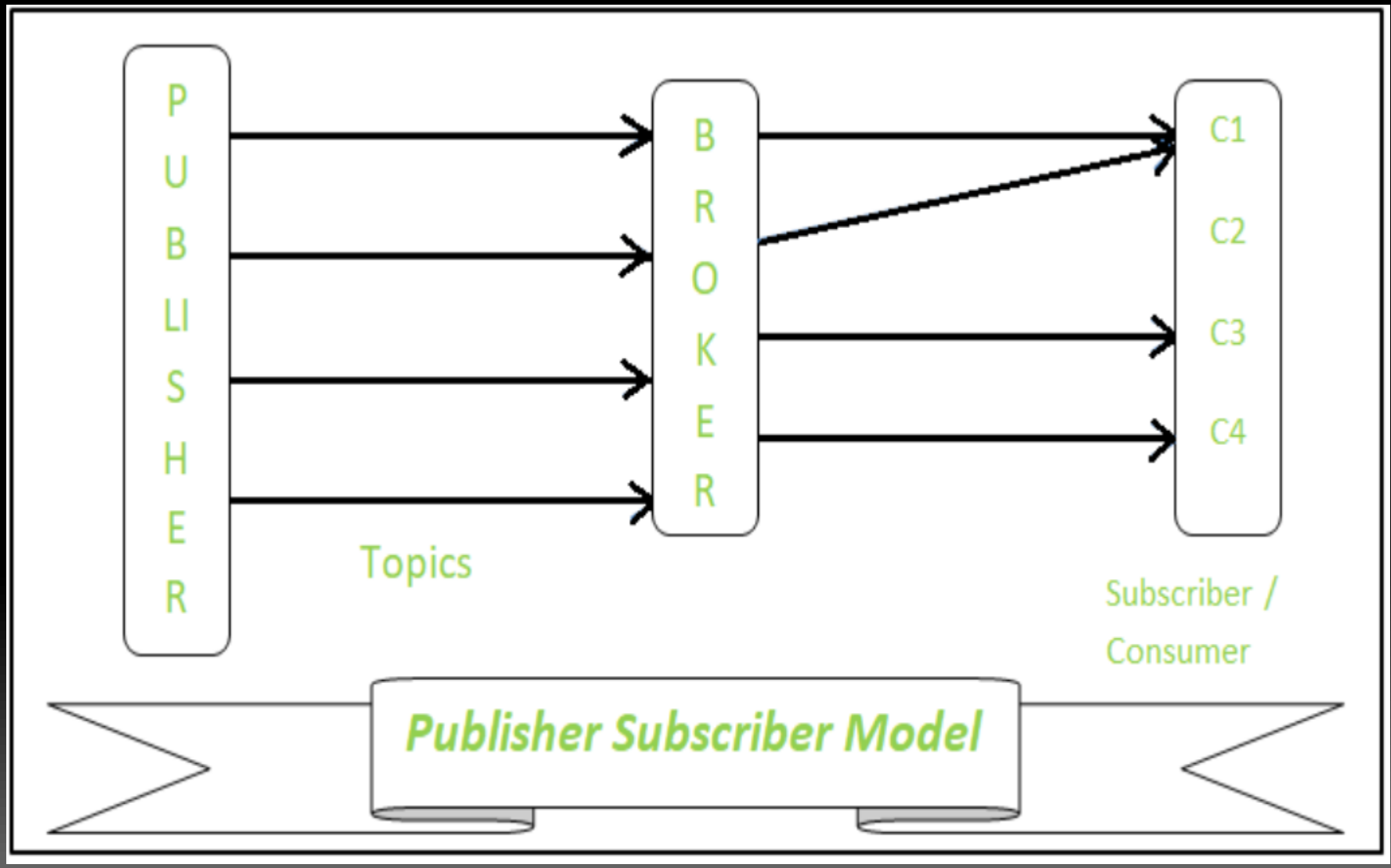
IoT Communicational Models and APIs

- IoT devices are found everywhere and will enable circulatory intelligence in the future. For operational perception, it is important and useful to understand how various IoT devices communicate with each other. Communication models used in IoT have great value. The IoTs allow people and things to be connected any time, any space, with anything and anyone, using any network and any service.

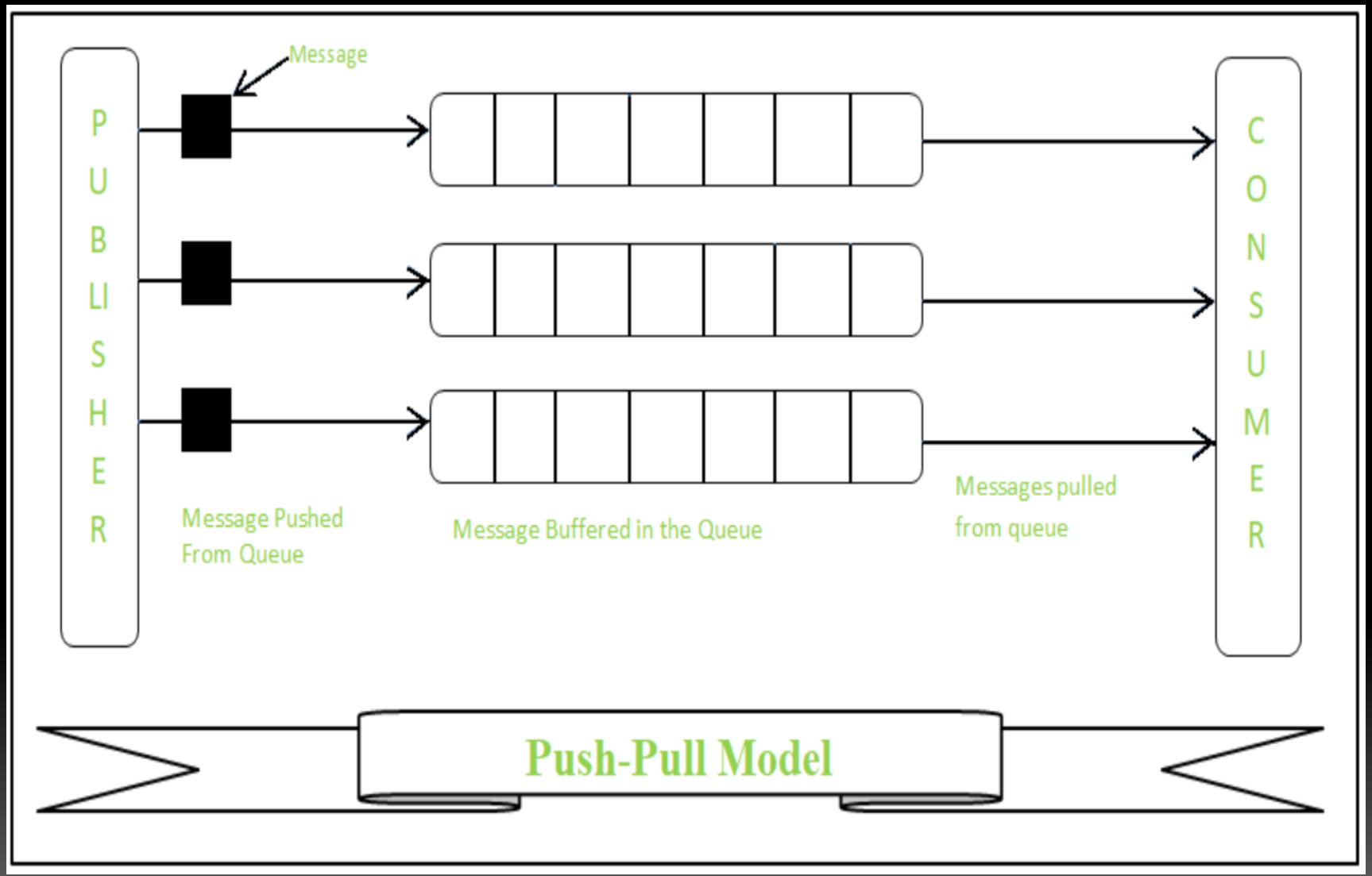
- **Types of Communication Model :**
- **1. Request & Response Model –**
  This model follows a client-server architecture.
- The **client**, when required, requests the information from the server. This request is usually in the encoded format.
- This model is stateless since the data between the requests is not retained and each request is independently handled.
- The server Categories the request, and fetches the data from the database and its resource representation. This data is converted to response and is transferred in an encoded format to the client. The client, in turn, receives the response.
- On the other hand — In **Request-Response** communication model client sends a request to the server and the server responds to the request. When the server receives the request it decides how to respond, fetches the data retrieves resources, and prepares the response, and sends it to the client.


Client
Request
Servers
DB
Fetch
Response
(Process &
Categories in
Server Section)
Process
Client
Response
Prepare Response (Encoded)
Request & Response Model

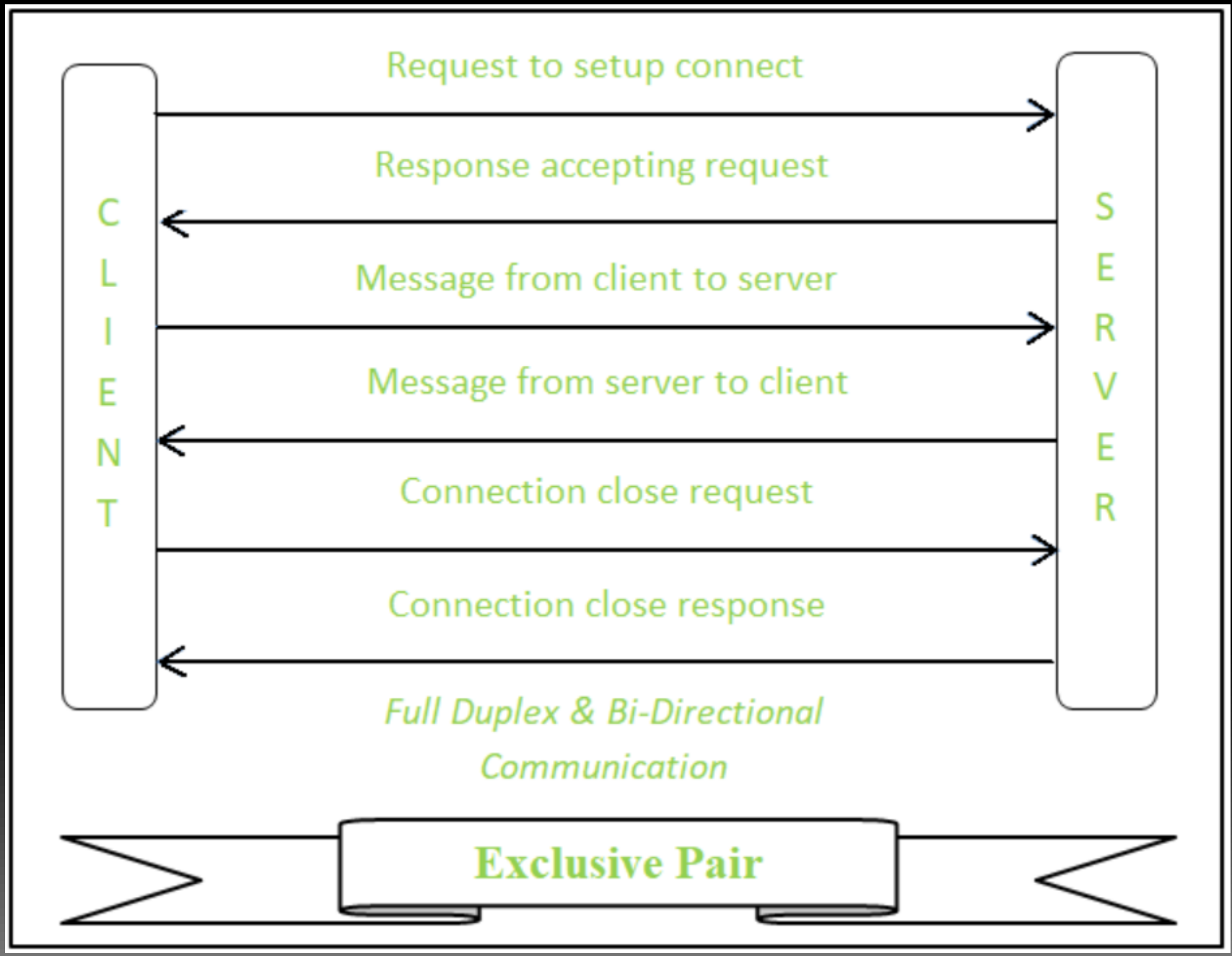
- **2. Publisher-Subscriber Model –**
  This model comprises three entities: Publishers, Brokers, and Consumers.
- **Publishers** are the source of data. It sends the data to the topic which are managed by the broker. They are not aware of consumers.
- **Consumers** subscribe to the topics which are managed by the broker.
- Hence, **Brokers** responsibility is to accept data from publishers and send it to the appropriate consumers. The broker only has the information regarding the consumer to which a particular topic belongs to which the publisher is unaware of.


PUBLISHER
BROKER
C1
C2
C3
C4
Topics
Subscriber / Consumer
Publisher Subscriber Model

- **3. Push-Pull Model –**
  The push-pull model constitutes data publishers, data consumers, and data queues.
- **Publishers** and **Consumers** are not aware of each other.
- Publishers publish the message/data and push it into the queue. The consumers, present on the other side, pull the data out of the queue. Thus, the queue acts as the buffer for the message when the difference occurs in the rate of push or pull of data on the side of a publisher and consumer.
- **Queues** help in decoupling the messaging between the producer and consumer. Queues also act as a buffer which helps in situations where there is a mismatch between the rate at which the producers push the data and consumers pull the data.


Message
PUBLISHER
CONSUMER
Message Pushed From Queue
Message Buffered in the Queue
Messages pulled from queue
Push-Pull Model

- **4. Exclusive Pair –**
- **Exclusive Pair** is the bi-directional model, including full-duplex communication among client and server. The connection is constant and remains open till the client sends a request to close the connection.
- The **Server** has the record of all the connections which has been opened.
- This is a state-full connection model and the server is aware of all open connections.
- Web Socket based communication API is fully based on this model.


Request to setup connect
Response accepting request
Message from client to server
Message from server to client
Connection close request
Connection close response
CLIENT
SERVER
Full Duplex & Bi-Directional Communication
Exclusive Pair